

**AN ARCHITECTURE FOR FEDERATED VIDEO PROCESSING AND
ONLINE STREAMING**

RAZIB IQBAL

Thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
In partial fulfillment of the requirements
For the degree of

**DOCTORATE OF PHILOSOPHY (PH.D.)
IN
COMPUTER SCIENCE**

Ottawa-Carleton Institute for Computer Science
School of Information Technology and Engineering
University of Ottawa

© RAZIB IQBAL, OTTAWA, CANADA, 2010

AN ARCHITECTURE FOR FEDERATED VIDEO PROCESSING AND ONLINE STREAMING

ABSTRACT

Today's multimedia communication has been greatly shaped by the coexistence of a number of complementary as well as competing access, delivery, and consumption technologies. Rich media are now accessible via numerous multimedia enabled devices through a wide variety of network types. "*What is missing*" is so far a mechanism to ensure that multimedia users can receive different qualities of video proportional to their device capabilities and network conditions.

In this thesis, we propose an online adaptive video streaming concept which takes into account different issues related to the Peer-to-Peer (P2P) content distribution paradigm such as peers' unreliability, as well as pragmatic aspects like receivers' heterogeneity. Our proposal also envelops compressed-domain video adaptation, watermarking, and perceptual encryption schemes where we utilize the MPEG-21 generic Bitstream Syntax Description (gBSD) as a content metadata.

The proposed architecture aims at online video adaptation with streaming in the Application Layer Multicast (ALM) overlays to serve heterogeneous devices including small handhelds. In this proposed design, participating peers act as the stream source, adaptation engine, and perform the streaming tasks.

We implemented a 3-in-1 adaptation-watermarking-encryption system to evaluate the compressed-domain adaptation performance. For the adaptive streaming, simulation is used to manifest that our design is robust, reliable, and suitable for multi-participant real-time collaboration and real-life deployment. The adaptive streaming system performance is also validated with the results found from an analytical model.

ACKNOWLEDGMENTS

All praise is due to the Almighty who has bestowed upon me the strength and patience to complete my Ph.D. research.

I would like to give my utmost appreciation to my thesis supervisor Professor Shervin Shirmohammadi whose mentorship has guided me to the way of innovation, accomplishment, and success. It has been an honor to be his first graduate student. I earnestly thank Professor Shervin for his constant support, enthusiasm, motivation, and guidance during the tough pursuit of my graduate studies.

A big bow to all my colleagues at the DISCOVER Laboratory and the Faculty of Engineering who made it a friendly place to study and work. Very special thanks to Dewan Tanvir Ahmed and Behnoosh Hariri. Their familiarity with the ideas of Peer-to-Peer content delivery was very helpful during the early programming phase of this undertaking. I gladly remember the teamwork of Saurabh Ratti and Sharmeen Shahabuddin who worked hard to develop my brainchild to completion in different occasions and co-authoring those results in published articles.

My deepest gratitude goes to my father S.M. Shaidul Alam and mother Kamrunnessa Begum for their unflagging love and care throughout my life. I am equally indebted to my sister, Tuhina Tasnin Luna, who has always been one of my best counselors. I owe loving thanks to my wife, Tasfia Ahmed Iqbal. Her encouragement was in the end what made this thesis come to fruition.

Finally, I would like to express my heartfelt thanks to Aminul Islam, Faisal Deen Arif, and Rizwana Alamgir-Arif who have always extended their helping hands here in Ottawa, Canada.

TABLE OF CONTENTS

ABSTRACT	II
ACKNOWLEDGMENTS	III
TABLE OF CONTENTS	IV
LIST OF FIGURES	IX
LIST OF TABLES	XI
LIST OF ABBREVIATIONS.....	XII
THESIS OUTLINE	XIV
CHAPTER 1.....	1
Introduction	1
1.1. Motivation	2
1.2. Associated Problems	5
1.3. Research Objectives	8
1.4. Research Contributions	10
1.5. Scholastic Achievements	12
CHAPTER 2.....	15
Background	15
2.1. Video Adaptation.....	15
2.1.1. Spatial Adaptation	16
2.1.2. Temporal Adaptation	17
2.2. MPEG-21 Digital Item Adaptation.....	17

2.3.	H.264 Video.....	20
2.4.	Integer Linear Programming	21
2.5.	Video Watermarking and Perceptual Encryption.....	23
CHAPTER 3.....		25
	Current State of Knowledge and Related Work	25
3.1.	Video Adaptation.....	25
3.2.	Video Watermarking and Encryption.....	28
3.3.	P2P Streaming	30
3.4.	Video Streaming.....	31
3.5.	Adaptive Video Streaming	32
CHAPTER 4.....		35
	Compressed-domain Video Adaptation, Authentication, and Encryption	35
4.1.	Compressed-domain Video Adaptation.....	35
4.1.1.	Temporal Adaptation	38
4.1.2.	Spatial Adaptation	39
4.1.3.	gBSD-based Adaptation of Continuous Streams	41
4.2.	Compressed-domain Video Authentication and Encryption.....	42
4.2.1.	Video Authentication	43
4.2.2.	Video Encryption.....	49
4.3.	Performance and Results.....	52
4.3.1.	Experiment Results for Compressed-domain Video Adaptation	52

4.3.2.	Experiment Results for Compressed-domain Video Authentication and Encryption	53
CHAPTER 5	56
	Proposed Federated Video Adaptation and Streaming Approach.....	56
5.1.	Adaptive Video Streaming	56
5.1.1.	Selecting the Number of Parents to Serve a Node	59
5.2.	Video Preparation and Buffering.....	61
5.2.1.	Processing the Video.....	61
5.2.2.	Video Buffering.....	62
5.3.	Overlay Construction.....	65
5.3.1.	The Tree Controller	66
5.3.2.	Peer Registration	67
5.3.3.	Parent Selection.....	69
5.3.4.	Parent Synchronization	69
5.3.5.	Peer Joining Constraints.....	71
5.3.6.	Peer Departure	72
5.3.7.	Tree Refinement.....	74
CHAPTER 6	76
	Adaptive Video Streaming Application Design and Performance Analysis.....	76
6.1.	Dynamic Quality Adjustment.....	78
6.2.	Service Fairness	79

6.3.	Peer Contribution	80
6.4.	Handling Network Fluctuations and Packet Loss.....	82
6.5.	Detecting Peer Failure.....	82
6.6.	Evaluation and Results.....	82
6.6.1.	Performance Metrics and Experiment Setup	83
6.6.2.	Adaptation Performance for Different Peer Class	86
6.6.3.	Impact of Wealthier Nodes in the Overlay	87
6.6.4.	Resource availability and QoE	88
6.6.5.	New Node Joining Success Rate	90
6.6.6.	Resource Utilization	92
6.6.7.	Internal Node Joining Success Rate	93
6.6.8.	Effectiveness of Minimum Contribute Rate.....	94
6.6.9.	Effect of the Dynamic Quality Adjustment Scheme	94
6.6.10.	Performance of Tree Refinement.....	95
CHAPTER 7	97
	Finding an Optimum Overlay	97
7.1.	Problem Formulation.....	97
7.2.	Revised ILP Model.....	101
7.3.	Finding the Solution	103
7.4.	Validation of the Heuristic Approach.....	104
CHAPTER 8	108

Conclusion.....	108
8.1. Federated Video Processing and Online Streaming.....	108
8.2. General Discussion.....	109
8.3. Future work.....	111
BIBLIOGRAPHY	112

LIST OF FIGURES

Figure 1. Conventional video distribution over the Internet	1
Figure 2. Universal Multimedia Access concept.....	4
Figure 3. Proposed P2P concept: No dedicated adaptation or media server	9
Figure 4. MPEG-21 digital item adaptation	18
Figure 5. gBSD-based adaptation architecture.....	36
Figure 6. gBSD transformation for Temporal Adaptation.....	38
Figure 7. Slicing strategies	40
Figure 8. Sequence diagram of a live video request.....	42
Figure 9. Proposed authentication architecture	44
Figure 10. Marking space selection from gBSD	45
Figure 11. A. Computing the hash value, B. Embedding authentication bits in slice	46
Figure 12. Compressed-domain encryption using gBSD.....	49
Figure 13. Spatial Adaptation resilient encryption.....	50
Figure 14. Sample video output after encryption	51
Figure 15. Performance of the 3-in-1 System for Pre-recorded Videos	55
Figure 16. Adaptive Video Streaming.....	57
Figure 17. Parent selection and Frameset dissemination in the multicast tree	60
Figure 18. Preparing video for temporal adaptation and streaming	62
Figure 19. Video Buffering, Adaptation, and Playing in live Adaptive streaming.....	63
Figure 20. Sliding window for Simultaneous Adaptation and Forwarding	64

Figure 21. gBSD transmission with video	64
Figure 22. Overlay construction: Solid lines and Dotted lines refers to Primary sender and Secondary sender, respectively.....	65
Figure 23. Illustration of adaptation and forwarding operations.....	70
Figure 24. Recovery from sudden node departure.....	72
Figure 25. Frameset skip due to sudden departure of a parent	73
Figure 26. Multi-parent Adaptive Video Streaming design - Solid lines and Dotted lines, refers to Video data and Control data, respectively	76
Figure 27. BSI for different number of parents.....	89
Figure 28. CSI for different number of parents	89
Figure 29. SDI for different number of parents.....	90
Figure 30. New node joining success rate.....	91
Figure 31. Percentage of total bandwidth usage.....	91
Figure 32. Percentage of total adaptation slots usage	92
Figure 33. Effect of Churn: Internal node joining success rate.....	93
Figure 34. Impact of the contribution rate	94
Figure 35. Tree refinement performance	95
Figure 36. Connection Matrix, P	98
Figure 37. Bandwidth utilization in single parent approach - Optimum vs. Proposed.....	106
Figure 38. Adaptation slot utilization in single parent approach - Optimum vs. Proposed.....	106

LIST OF TABLES

Table 1. Sample vlc_{n_T} bits after adaptation	47
Table 2. Digital Item generation performance.....	53
Table 3. Spatiotemporal adaptation performance: Proposed vs. Cascaded	53
Table 4. Parent selection	60
Table 5. Different system components of the adaptive video streaming system.....	77
Table 6. gBSD-based spatiotemporal adaptation performance for different peer class	87
Table 7. Impact of wealthier peers' availability on successful overlay generation.....	88
Table 8. Summary of notations.....	99
Table 9. Validation of the multi-parent adaptive streaming system: Optimum vs. Heuristic.....	107

LIST OF ABBREVIATIONS

Abbreviation or Symbol	Term
3G	Third Generation
ALM	Application Layer Multicast
AVC	Advanced Video Coding
BSD	Bitstream Syntax Description
B-Frame	Bi-directional Frame
CIF	Common Intermediate Format
DCT	Discrete Cosine Transform
DHT	Distributed Hash Table
DOM	Document Object Model
DRM	Digital Rights Management
DWT	Discrete Wavelet Transform
DI	Digital Item
FPS	Frames Per Second
gBSD	Generic Bitstream Syntax Description
GPRS	Global Packet Radio Services
HD	High Definition
I-Frame	Intra Frame
ILP	Integer Linear Programming
P2P	Peer-to-Peer
PDA	Personal Digital Assistant
PDP	Peer Discovery Protocol

P-Frame	Inter Frame
QCIF	Quarter Common Intermediate Format
QoE	Quality of Experience
ROI	Region of Interest
SD	Standard Definition
SVC	Scalable Video Coding
SQCIF	Sub Quarter Common Intermediate Format
UMA	Universal Multimedia Access
XML	Extensible Markup Language
XSL	Extensible Stylesheet Language
XSLT	XSL Transformations

THESIS OUTLINE

- Chapter 1** The scope of this thesis is defined as well as motivation, problems related to this research, research objective, and specific contributions are highlighted in Chapter-1.
- Chapter 2** Fundamental concepts associated with this research along with the motivation for utilizing H.264 video format and MPEG-21 gBSD as content metadata are briefly discussed in Chapter-2.
- Chapter 3** Some significant related work covering online video adaptation, authentication, encryption, P2P streaming, peer contribution incentive, and analytical model for online video distribution are highlighted in Chapter-3.
- Chapter 4** The metadata generation and compressed-domain video processing steps are detailed in Chapter-4.
- Chapter 5** Design of the proposed simultaneous adaptation and streaming model in a P2P environment is presented in Chapter-5. First, how the gBSD-based adaptation can be applied to P2P streaming is highlighted, and then overlay formation basics are introduced.
- Chapter 6** Different essential components of a multi-parent adaptive video streaming system as well as overlay service fairness, minimum contribution requirement by the peers, and a dynamic quality adjustment scheme are presented in Chapter-6.
- Chapter 7** A mathematical model based on the Integer Linear Programming (ILP) problem to find the optimal overlay streaming performance is described in this chapter. A basic model captures the single parent based overlay streaming. A revised model incorporates multiple parent based overlay streaming including service

fairness, minimum resource contribution, and dynamic quality adjustment constraints.

Chapter 8 Some thoughts pertaining to the overall design is presented in this last chapter. The thesis ends with some concluding remarks and an insight of what can be investigated in the future.

CHAPTER 1

INTRODUCTION

In the modern world, people are no longer satisfied with just sound or still images for their day-to-day communication and entertainment needs. Motion picture has become a means of conveying expression and information from one end to another. Video telephony, video conferencing, video streaming are just a few of many well-known terms which are broadly related to different video applications. The enthusiasm of accessing online media contents has increased due to the fact that rich media transmissions and Internet access now have become possible ubiquitously through the advances in mobile 3G and beyond 3G networks. Portable entertainment systems, cell phones, and PDAs are now able to handle multiple applications - music, games, photo, video and wireless connectivity - with Internet as a key feature. However, limitations and variations of these devices make it difficult to render and display regular videos on these devices like that of desktops or laptops. Usually, commercial video distribution systems serve customers with dedicated media servers via high-speed Internet or cable TV. Each service provider addresses requirements of the subscribers with a unique edge and proprietary delivery model to offer a negotiated quality of viewing experience. In some cases, adaptation servers are deployed to adapt the video quality in order to meet the end-user requirements as shown in Figure 1.

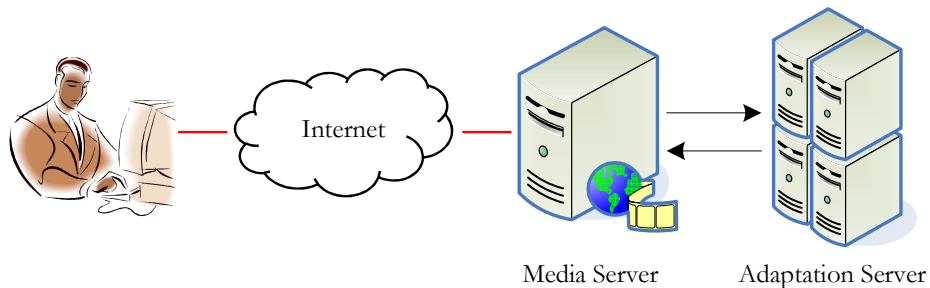


FIGURE 1. CONVENTIONAL VIDEO DISTRIBUTION OVER THE INTERNET

Alternatively, online video streaming has gained huge popularity with a very high number of active participants. Video sharing websites like YouTube, Yahoo! Video are offering free services whereby users can upload videos and share them openly. ComScore [1] reports that in May 2007 nearly 75% of Internet users watched an average of 158 minutes of online video during the month, where 3 out of 4 Internet users streamed video over the Internet in the United States. Alongside, Peer-to-Peer (P2P) content distribution and sharing concept has fascinated researchers due to its ability to disseminate large amount of data at a reduced deployment cost. Because of its popularity and success, lately, there has been a significant interest in the use of the P2P concept in real-world applications over the Internet like file sharing and video streaming. A reasonable number of P2P video communities are now formed in practice, as evident by various application groups such as PPMate, SopCast, TVUPlayer, and TvAnts¹. According to Setton et al. [2], nowadays, P2P video streaming represents more than 60% of the total internet traffic. While still not of the highest quality, these platforms allow for reasonable live P2P video broadcasting of sporting events and TV channels in a community with a large number of global users.

1.1. Motivation

Traditional dedicated media adaptation and streaming server based content delivery architectures over the Internet usually suffer from high workload and capacity constraints. The high cost of bandwidth and maintenance required for server-based solutions are some factors that undesirably allow only limited resource providers to broadcast media rich contents to end-users over the Internet. For example, the bandwidth provisioning cost of popular YouTube servers are estimated at more than one million dollar per day [3]. IP multicasting [4] attempted to deal with this problem by conserving resources in the routers or by load balancing across a large number of edge servers [5]. Even then, the problem of scalability in rich media distribution and streaming to a large number of simultaneous users is only mitigated to a certain degree, not

¹ www.ppmate.com, www.sopcast.org, www.tvunetworks.com, tvants.en.softonic.com

solved. Therefore, we need an efficient and practical technique for robust and scalable distribution of multimedia contents to satisfy large groups of users' needs endowed with limited network/computing resources.

Another related issue is the heterogeneity in multimedia enabled devices. For example, mobile phones typically have a smaller screen and lower bandwidth via Global Packet Radio Services (GPRS), while PDAs have higher resolution screens and can have access to broadband connections via Wi-Fi. In any case, modern smart handheld devices are geared towards accessing on-line multimedia resources and video streams. Online video sites like YouTube, DailyMotion, Vimeo², and several other video portals have been offering High-Definition (HD) videos to public viewers for a while. However, the unavoidable limitations of small handhelds (e.g. limited memory, computing power, battery life, bandwidth, etc.) restrict them to become fully functional autonomous nodes in an online application. There also exist intricacies to render even Standard-Definition (SD) videos to multimedia-enabled handheld devices like that of laptops or desktops. A commonly used solution today is to store multiple versions of the same content taking into consideration different network types and device capabilities. Apparently, this is not a scalable solution given the numerous device types entering the market each day. Therefore, in order to reduce complexity and to support the concept of Universal Multimedia Access (UMA), it is desired to have an adaptation system that is generic and format independent. It will also allow service providers to target a wide range of devices and networks, maximize their customers' experience, and minimize storage and maintenance requirements on the server side.

As mentioned earlier, the interest in the P2P concept lies in its ability to distribute large amounts of data at a reduced deployment cost. The basic concept behind P2P streaming is that peers forward the video stream to each other while also viewing it. Surprisingly, despite its high prospect, P2P video streaming systems seem to be in their early stages compared to other P2P applications such as file sharing. The reason behind this slow progress is that P2P video

² www.youtube.com, www.dailymotion.com, www.vimeo.com

applications are much more challenging as they not only need to support a large number of participants simultaneously, but also need to deal with each participant's dynamic changes to assorted device requirements and varying bandwidth.

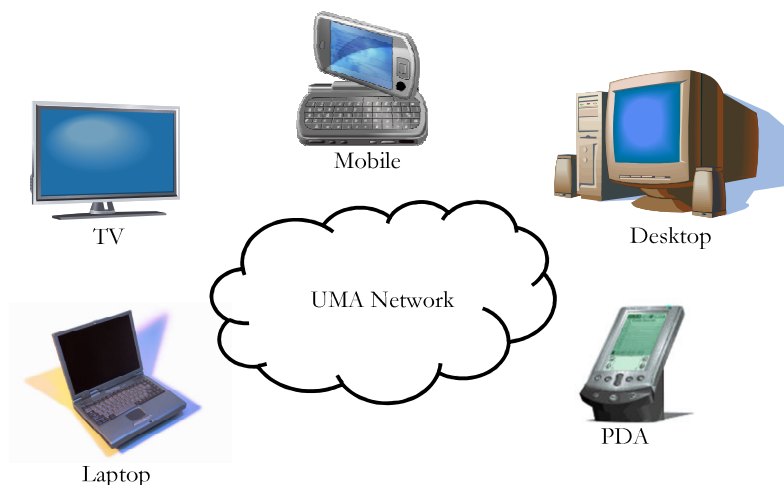


FIGURE 2. UNIVERSAL MULTIMEDIA ACCESS CONCEPT

Considering the above issues, it is essential to have an architecture so that different multimedia capable devices have access to different video contents in an interoperable manner providing their own set of distinct combination of features. The motivation comes from the perception that with the modern computing capabilities, a peer may carry out sufficient adaptation operations and stream media contents to other peers in the presence of some incentives. In fact, personal computer based hubs can bring a tremendous revolution with increased processing power combined with Internet connections and connectivity to a variety of displays. By distributing the workload to low-cost, off-the-shelve computing hosts such as personal computers and workstations, one can eliminate the need for costly centralized adaptation and streaming servers, and at the same time, improve the system's scalability and availability. Therefore, in this thesis, the focus is on designing a cooperative video streaming system, based on the P2P content distribution concept, to simultaneously adapt and stream video

contents to heterogeneous users. It leads to an efficient application side P2P video streaming service over the Internet addressing the UMA concept as shown in Figure 2.

1.2. Associated Problems

Despite a significant interest in the P2P technologies, there is a lack of a scalable design for supporting mobile and heterogeneous devices in the P2P systems. In this section, some of the associated problems that are the reason for this lacking are highlighted.

- In online streaming, different recipients may experience different bandwidth and device constraints. Since the knowledge of the participating peers is not known a priori, the stream-source must choose the video quality and size in advance. However, depending on the bandwidth/device capabilities of the participating peers, the selected quality may be too high or too low. If the rate is too high, then the incoming peers may face denial of service. Yet again, if the rate is too low, then the system resource in terms of upload-download capacity will not be fully exploited. Therefore, the architecture should provide every intended recipient with the required quality that it could sustain. For example, in a simple scenario, temporal adaptation is necessary to meet the diversity of network types that requires flexible media contents to fit network pipes of different bandwidth. However, diverse network types, device variations, user preferences, and video codec types lead to manifold of adaptation requirements. An easy way out to meet the device diversity is to design the streaming system for the worst-case scenario. Another option is to produce and store content in several formats while taking into account a wide variety of possible user devices and preferences, and making an appropriate selection at the delivery time. The drawback of this approach is that recipients with higher capability maybe deprived from what they can handle, which will eventually limit the system's ability to adapt dynamically.

- Video content delivery frameworks differ mainly due to pre-encoded contents and live-video contents. From the application perspective, these frameworks can be further categorized into the following two classes - Online applications (e.g. file sharing and/or file download), and Real-time applications (e.g. video conferencing and/or live streaming). Now, to address the issue of serving heterogeneous devices, real-time adaptation is a big challenge. A straightforward solution is to perform cascaded decoding-transcoding-reencoding operations in some intermediary nodes between the content provider and the recipient. However, it requires additional processing power and processing time. Within a P2P system, placing a dedicated server to carry out necessary adaptation operations may not be feasible for real-time adaptation and delivery of live video streams. Moreover, such dedicated servers may cause service bottlenecks.
- Conventionally, video metadata describe the events and objects occurring in the video scenes, creator identification, publisher, etc. Compared to the conventional ones, metadata describing syntax and semantics of the video codec is relatively a new concept. Without a standardized metadata support, a variety of adaptation operations in a format agnostic way within a distributed multimedia environment is not easy. Additionally, to remove the incompatibility of the metadata description for different video codecs, it is necessary to have a standard metadata description format.
- Although in theory, IP-multicast is the most efficient multicasting vehicle for Internet video broadcasting/streaming, its deployment remains limited due to the unsolvable technical, management, and business issues, which have prevented service providers to give multicast support to the Internet users at homes. As remedies to the unavailability of IP multicasting, research pioneers have been focusing on Application Layer Multicasting (ALM) for at least the past ten years [6]. One of the advantages of ALM is the flexibility to use multiple spanning trees simultaneously to improve the achievable throughput. In literature, excellent research entailing ALM-based content distribution architectures include NICE [7], ZIGZAG [8], and Splitstream [9] to name a few. Most of these

architectures, however, focus on protocol design and do not consider live adaptation of the contents to meet user heterogeneity.

- For adaptive video streaming, we need to determine the video quality to be delivered to each user. Adaptation decision taking in a dynamic streaming environment mainly depends on the actual resource characteristics, e.g. required bitrate, and the usage environment, e.g. available bandwidth, both of which may vary between video stream fragments during a streaming session. Ordinary P2P video broadcasting communities that are now deployed in practice allow only one fixed video stream configuration, which eventually makes the participation of heterogeneous devices uncertain. Certainly, it is unfair to allocate the same video quality to all peers.
- Another key issue is the participant's quality of experience (QoE). The streaming protocol has to operate well in adversarial scenarios such as frequent node failures, rapid node joining and leaving (also known as churn), and uncooperative peers in order to offer an uninterrupted viewing experience. Additionally, in order to enable more users to join a streaming session at any instant, there needs to be a mechanism to create a balance between the QoE and the service ratio.
- Recently, researchers have proposed a few promising solutions to address the need of adaptive video streaming. However, it is hard to judge the efficiency of these proposals when there is no reference that can be considered as the optimal boundary. Even though some of the existing analytical models for P2P streaming (e.g. [10][11]) consider system level design issues, so far none of the studies have formulated a tractable analytical model to find the optimum overlay considering resource utilization and peer heterogeneity.

Based on the above observations, we conclude that a research initiative is needed to investigate the possibility of incorporating online, distributed, and real-time video adaptation mechanism to the P2P video streaming concept in order to make it robust and more attractive to the Internet community.

1.3. Research Objectives

A basic problem in online media distribution is how to generate an optimum routing structure that delivers the media content from sender(s) to receivers, thus achieving a certain optimization objective such as resource utilization and fairness. Adding to this complexity of the problem is the heterogeneity challenge. As users connect to the Internet via a variety of access technologies, with an order-of-magnitude difference between inbound and outbound capacities, a one-size-fits-all streaming quality simply cannot be found. The remedy to this challenge is the real-time adaptation of videos, in which the content will be adapted according to the device or network requirements. Since adapting the media contents according to users' context sets the primary challenge towards seamless access in a ubiquitous computing environment, in this thesis an adaptive multimedia sharing architecture is proposed. Our objective in this thesis is to enable a community-driven federated video adaptation and sharing concept. We propose an architecture, which adapts video data online in the compressed-domain and within the participating nodes in an ALM overlay. In order to provide a better understanding of the concept, consider the following application scenario:

A mobile user wants to watch a live soccer broadcast through a service such as SopCast or PPStream. The service works fine for most PC users with high-speed Internet connection. However, the video resolution is too large for the mobile device's screen, and the offered bit rate is too high for the wireless connection. As such, most mobile users are locked out of the system. It would be a significant contribution if the system would not only use peers to stream the video among themselves, but also use some of those peers to adapt the video to the capabilities of some other peers.

In the above example, in order to match the screen size of the handheld, other users (a.k.a. peers) need to adapt the video stream spatially. Moreover, due to the mobility of the handheld user, bandwidth may also vary over time. Therefore, the frame rate needs to be adapted accordingly in order to maintain good audio-visual quality. Consequently, the spare computing power (i.e. CPU) of the peers can be used to serve other peers who require adapted videos. This

way we can eliminate the need for dedicated media streaming and/or adaptation servers as illustrated in Figure 3. It also gives heterogeneous devices including small handhelds the opportunity to participate in the video streaming systems.

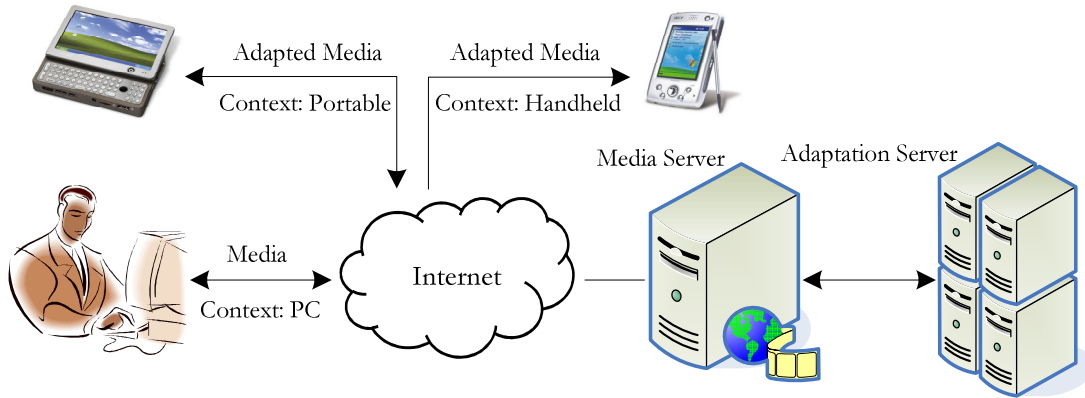


Figure 3. Proposed P2P concept: No dedicated adaptation or media server

Now, a cooperative video streaming system needs to be contribution-aware as the contributions from the peer nodes are most likely to be heterogeneous in the real world. To facilitate adaptive video streaming, peers need to be reluctant in terms of sharing not only their bandwidth, but also their computing power to simultaneously adapt and stream multimedia contents. To utilize the peer resources in order to carry out the necessary adaptation and streaming tasks, a mechanism is needed that will motivate participating peers to contribute their available resources. Without such a mechanism, peers may free ride or contribute at low rates. For contribution awareness, we need to establish meaningful charging models and incentive mechanisms. Moreover, computing an optimal overlay heuristically which conforms to the incentives of the peers and deals with all the system uncertainties would be unattainable because of the volatile nature of the streaming environment and participants. An analytical model of the system, therefore, will quantify the expected performance at a given time stamp. Such an analytical model usually helps to evaluate the overall system performance, relationships of important system parameters, and metrics of the heuristic approach.

Simultaneously, security and Digital Right Management (DRM) of multimedia data has become a major concern, especially in UMA networks. Conventional video authentication techniques do not allow any form of manipulation of the encoded bitstream, thus making it impractical for adaptation operations. As a result, an authentication and perceptual encryption scheme is needed to assure integrity of sensitive video contents as an embedded feature of the adaptation practices.

In a nutshell, the research that was performed during the course of this doctoral studies investigated a novel video streaming and distribution architecture that introduces an online adaptation and streaming mechanism taking into consideration the peer heterogeneity, DRM, and optimum overlay computation.

1.4. Research Contributions

In this thesis, a novel online adaptive video streaming concept is introduced which takes into account different issues related to the P2P paradigm such as peers' unreliability, as well as pragmatic aspects like receivers' heterogeneity. Additionally, basic principles of a joint adaptation, authentication, and encryption scheme for H.264 video are also proposed. Therefore, major contributions are in the area of compressed-domain adaptation, watermarking, perceptual encryption, and streaming of the H.264 videos using P2P content distribution concept. Specific contributions are listed below:

- **Compressed-domain Adaptation.** To avoid server-based video adaptation, we propose a lightweight H.264 video adaptation scheme. The proposed adaptation system performs spatiotemporal adaptation operations on ordinary personal computers. Both spatial and temporal adaptations were evaluated to see the real-time performance of the proposed compressed-domain adaptation mechanism.

- **Video Authentication.** We propose a digital watermarking based authentication scheme for H.264 video. The scheme can be operated in an intermediary node along with the adaptation operations.
- **Video Encryption.** A scheme for compressed-domain spatiotemporal adaptation resilient perceptual encryption of H.264 videos is also presented. The encryption scheme partially degrades the visual quality of the video content in order to restrict the full quality to legitimate users only.
- **Adaptive Video Streaming.** A novel P2P streaming architecture is designed which supports the simultaneous adaptation and streaming of video contents to heterogeneous peers. The adaptive video streaming architecture utilizes content metadata and compressed-domain video processing techniques to meet the real-time video adaptation needs.
- **Analytical Model.** A mathematical model, which is based on the Integer Linear Programming (ILP) method, is developed for the adaptive video streaming design. The model captures the essential elements of the adaptive video streaming system as well as computes the optimum solution at a given time. The model also illustrates the relationship among all parameters that affect the efficiency of streaming, and computes the trade-offs that exist between service fairness and system efficiency. Finally, it helps to evaluate and compare the overall performance of the video distribution system by means of total resource utilization in a streaming overlay.
- **Validation and Performance.** Compressed-domain adaptation performance is evaluated from the implemented 3-in-1 adaptation-watermarking-encryption system. A traditional cascaded adaptation mechanism is also implemented to compare with the proposed codec-agnostic adaptation mechanism. For the adaptive streaming, simulation is used to manifest that the design is robust, reliable, and suitable for

multi-participant real-time collaboration and real-life deployment. The overall system performance is validated with the results found from the analytical model.

1.5. Scholastic Achievements

With respect to the research objectives and research contributions described above, overall, this research has resulted in 12 papers published in conference proceedings, 2 journal papers, 2 book chapters, and 3 demonstrations. A complete list of scholastic achievements and publications are listed below:

Journal Papers

- [1] R. Iqbal and S. Shirmohammadi. DAg-stream: Distributed Video Adaptation for Overlay Streaming to Heterogeneous Devices. In Springer's Peer-to-Peer Networking and Applications, vol. 2, no. 3, pp. 202-216, 2009.
- [2] R. Iqbal, S. Shirmohammadi, A. El Saddik, and J. Zhao. Compressed Domain Video Processing for Adaptation, Encryption, and Authentication. In IEEE Multimedia, vol. 15, no. 2, pp. 38-50, 2008.

Book Chapters

- [1] R. Iqbal and S. Shirmohammadi. Spatio-Temporal H.264/AVC Video Adaptation with MPEG-21. In Handbook of MPEG Applications: Standards in Practice, M. C. Angelides and H. Agius, John Wiley & Sons Ltd, ISBN: 978-0-470-75007-0, pp. 205-220, 2010.
- [2] R. Iqbal and S. Shirmohammadi. MPEG-21 Based Video Adaptation with Encryption and Authentication. In Encyclopedia of Multimedia, 2nd Edition, Borko Furht, Springer, ISBN: 978-0-387-74724-8, pp. 449–457, 2008.

Refereed Demos

- [1] R. Iqbal, S. Shahabuddin, and S. Shirmohammadi. Compressed-Domain Spatio-Temporal Adaptation System for Video Delivery. In ACM Multimedia, 2009.
- [2] R. Iqbal and S. Shirmohammadi. MPEG-21 Based Temporal Video Adaptation for Heterogeneous Devices and Mobile Environments. In IEEE ICME, 2009.

- [3] R. Iqbal and S. Shirmohammadi. MPEG-21 Adaptation of Video Learning Objects. In I²LOR, 2006.

Conference Papers

- [1] R. Iqbal and S. Shirmohammadi. An Analytical Approach to Model Adaptive Video Streaming and Delivery. In Proceedings of Workshop on Advanced Video Streaming Techniques for Peer-To-Peer Networks and Social Networking (AVSTP2P), in conjunction with ACM Multimedia, pp. 55-58, 2010.
- [2] R. Iqbal, S. Shahabuddin, and S. Shirmohammadi. Compressed-Domain Spatial Adaptation Resilient Perceptual Encryption of Live H.264 Video. In Proceedings of International Conference on Information Science, Signal Processing and Their Applications (ISSPA), pp. 472-475, 2010.
- [3] S. Shahabuddin, R. Iqbal, A. Nazari, and S. Shirmohammadi. Compressed Domain Spatial Adaptation for H.264 Video. In Proceedings of ACM Multimedia, pp. 797-800, 2009.
- [4] R. Iqbal, S. Ratti, and S. Shirmohammadi. A Distributed Camera Network Architecture Supporting Video Adaptation. In Proceedings of ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC), pp. 1-7, 2009.
- [5] R. Iqbal and S. Shirmohammadi. A Light-Weight Federated Video Adaptation System for P2P Overlays. In Proceedings of IEEE International Conference on Multimedia and Expo (ICME), pp. 1126-1129, 2009.
- [6] R. Iqbal and S. Shirmohammadi. Cooperative Video Adaptation And Streaming Scheme for Mobile and Heterogeneous Devices in a Community Network. In Proceedings of IEEE Workshop on Community Driven Mobile Multimedia, in conjunction with IEEE ICME, pp. 1768-1771, 2009.
- [7] S. Shahabuddin, R. Iqbal, S. Shirmohammadi, and J. Zhao. Compressed-Domain Temporal Adaptation-Resilient Watermarking For H.264 Video Authentication. In Proceedings of IEEE Workshop on Multimedia Security and Content Protection, in conjunction with IEEE ICME, pp. 1752-1755, 2009.
- [8] R. Iqbal and S. Shirmohammadi. Online Adaptation for Video Sharing Applications. In Proceedings of ACM Multimedia, pp. 853-856, 2008.
- [9] R. Iqbal, B. Hariri, and S. Shirmohammadi. Modeling and Evaluation of Overlay Generation Problem for Peer-assisted Video Adaptation and Streaming. In Proceedings

of ACM International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV), pp. 87-92, 2008.

- [10] R. Iqbal, D.T. Ahmed, and S. Shirmohammadi. Distributed Video Adaptation and Streaming for Heterogeneous Devices. In Proceedings of IEEE Workshop on Mobile Peer-to-Peer Computing, in conjunction with IEEE International Conference on Pervasive Computing and Communications (PerCom), pp.492-497, 2008.
- [11] R. Iqbal, S. Shirmohammadi, and J. Zhao. Compressed Domain Authentication of Live Video. In Proceedings of IEEE International Conference on Signal Processing and Communication (ICSPC), pp. 1443-1446, 2007.
- [12] R. Iqbal, S. Shirmohammadi, and J. Zhao. Hard Authentication of H.264 Video Applying MPEG-21 Generic Bitstream Syntax Description (GBSD). In Proceedings of IEEE International Conference on Multimedia and Expo (ICME), pp. 875 – 878, 2007.

Prizes and Awards

- [1] OCRI Futures Award: Student Researcher of the Year. Ottawa, Canada, 2010.
- [2] ACM Multimedia student conference participation grant. Beijing, China, 2009.
- [3] 1st prize. R. Iqbal and S. Shirmohammadi. Online Adaptation for Video Sharing Applications. uOttawa Graduate Studies & Research Day Poster Competition, 2009.
- [4] 2nd prize. R. Iqbal, S. Shirmohammadi, and J. Zhao. Hard Authentication of H.264 Video Applying MPEG-21 GBSD. uOttawa Graduate Studies & Research Day Poster Competition, 2008.
- [5] 3rd Prize. R. Iqbal and S. Shirmohammadi. Adaptation of Video Learning Objects Using MPEG-21 DIA. I²LOR Demo Competition. In Annual Scientific Conference of LORNET Research Network, Montreal, Canada, 2006.

CHAPTER 2

BACKGROUND

In this thesis, a federated video adaptation and streaming architecture based on the P2P concept is portrayed. The notion behind P2P video streaming is to forward the video stream to other users while also viewing it in order to avoid (or reduce) the deployment of costly video streaming servers. In P2P streaming, participants do not have to get the media content from the original media source. Instead, a peer connects to one or more of its neighboring peers, which are currently tuning in to receive the same media stream. Recursively, the neighbors are connected all the way up to the original media source node. Now, the novelty of the proposed architecture is the addition of an adaptation mechanism before streaming the video segments to another peer. Content metadata is utilized to identify the entities for adaptation, authentication, and encryption. Before we go into the further details of the architecture, basics of the video adaptation, content metadata format, H.264 video format, and Integer Linear Programming are briefly introduced in this chapter.

2.1. Video Adaptation

Video transcoding is the frequently applied technique for video customization. Video transcoding is defined as the conversion of digital video content from one format to another. It involves decoding/decompressing the original data to a raw intermediate format (i.e. YUV) first and then re-encoding this into the target format. Scalable or non-scalable transcoding approaches to adapt video refers to adapting various coding parameters such as frame rate, spatial resolution, or color. While homogeneous transcoding is done at the same coding standard, heterogeneous transcoding converts from one standard format to another. So, video adaptation can be denoted as homogenous video transcoding, which aims to reduce bit rate, frame rate, and/or the resolution of the pre-encoded video stream. It does not involve any kind of syntax modifications

to the coded video data. Therefore, the incoming compressed video stream preserves its format and compression characteristics after adaptation operation.

Adaptation operations are typically performed in the intermediate locations between the server and the client (e.g. proxy servers), although in some applications, the operations may be included in the servers or clients. Video bitstream entities are of different types and can be identified in different levels such as pixel, frame, group of pictures, etc. Different modules can be designed depending on the variety of these entities. For example, a video frame can be reduced in resolution or skipped in order to reduce the overall bandwidth.

2.1.1. Spatial Adaptation

The goal of spatial adaptation is to adapt video frames for a target resolution or screen size. Spatial adaptation can be done in two ways - either downscaling the video for a particular display resolution or cropping a particular region from the video stream. When cropping a desired region from the video, it is impractical to crop the video during the initial compression because it would require multiple encoded videos – a different one for each cropped region. Having the ability to spatially adapt an already compressed video will allow an intermediary node to serve different regions of video frames based on client demands from a single encoded high-resolution video stream. Cropping can be done in several ways. One approach is to transcode the video (i.e. to decompress the video, crop the desired regions, and then recompress it) before transmitting the video to the end user. While simple, this approach requires significant computational overhead, which makes it less suitable for real-time applications. Another approach is to create many smaller regions of the video and compress them separately. In the latter approach, it is easier to select particular regions from the video. However, it requires the client-side application to synchronize and merge multiple video streams for display, which eventually leads to added complexities.

2.1.2. Temporal Adaptation

Temporal adaptation allows meeting end user requirements when an end system supports only a lower frame rate due to limited processing power or low bandwidth availability. Frame dropping is a well-known technique for temporal adaptation. It tries to meet the desired frame rate by dropping or skipping frames in a video bitstream. A major concern in frame dropping is that if an anchor/reference frame is dropped, subsequent frames may need to be re-encoded to avoid error propagation. Moreover, when frames are skipped, the motion vectors cannot directly be reused because the motion vectors from the incoming video frame point to the immediately previous frame. If the previous frame is dropped during the adaptation process then the link between two frames is broken and the end decoder will not be able to reconstruct the picture using these motion vectors. To avoid such problems, some video codec offer multiple reference frames for motion compensation.

2.2. MPEG-21 Digital Item Adaptation

Without a standard metadata support, the variety and the incompatibility of adaptation approaches in a distributed environment makes multimedia adaptation procedures complicated. In the past, metadata aimed at describing the semantics of the content by means of keywords, violence ratings, or genre classifications. Metadata standards supporting this type include MPEG-7, TV Anytime, etc. ISO/IEC 21000 Multimedia Framework, known as MPEG-21, was initiated because of the lack of interoperability among advanced multimedia packaging and distribution applications. The MPEG-21 framework specifies a set of tools in order to describe the media content, the adaptation possibilities, and the usage context in the form of XML documents.

MPEG-21 Part-7 (Digital Item Adaptation, in short DIA) [12] provides normative description formats for the media object to enable interoperability. DIA specification details the syntax and semantics of tools that may be used to assist adaptation of a Digital Item (DI). A DI is referred to as a bitstream together with all its relevant descriptions. In simplified terms, the bitstream can be audio, video, or any other media. The advantage of MPEG-21 DIA is that it will

benefit all the stakeholders in the multimedia delivery and consumption chain (e.g. Service and Content Providers, Network Operators, Device Manufacturers, and End Users) in terms of interoperability and compatibility of products. This standard can be used in any application domain because it has been designed in a protocol- and application-independent way. An overview of DIA, its use in multimedia applications, and report on some of the ongoing activities in MPEG on extending DIA for use in rights governed environments, are available in the literature [13][14] and are beyond the objective of this research.

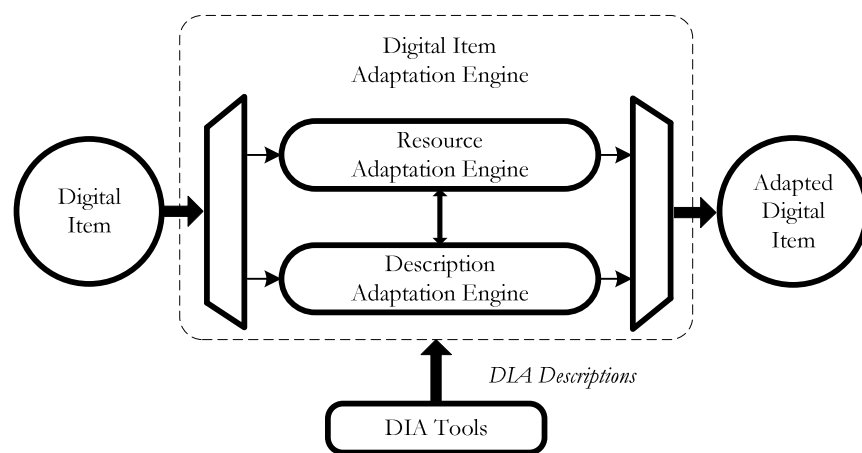


FIGURE 4. MPEG-21 DIGITAL ITEM ADAPTATION

In MPEG-21, coding format independence is accomplished by means of generic Bitstream Syntax Descriptions (gBSDs), and device interoperability is achieved through a unified description model that covers information about the user characteristics, terminal capabilities, network conditions, etc. This context information is generally referred to as the Usage Environment Description (UED). In the course of video adaptation, the gBSD of a video bitstream is transformed first, followed by the generation of the adapted bitstream from the original one, and guided by the modified gBSD. The concept of MPEG-21 DIA and the structure of the DIA engine are illustrated in Figure 4. From this figure, we can see that, a DI is

subject to a Resource Adaptation Engine and a Description Adaptation Engine, which together produce the adapted Digital Item.

BOX 1. SAMPLE GBSD REPRESENTATION

```
<dia:DIA
  xmlns="urn:mpeg:mpeg21:2003:01-DIA-gBSD-NS"
  xmlns:dia="urn:mpeg:mpeg21:2003:01-DIA-SpatialAdaptation-NS"
  xmlns:dia="urn:mpeg:mpeg21:2003:01-DIA-NS"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <dia:Description xsi:type="gBSDType">
    <Header><!-- ... and so on... --></Header>
    <gBSDUnit xsi:type="gbsdsa:gBSDFrameUnitType"
      syntacticalLabel=":I-Frame:0" start="576" length="34660"
      left="0" right="352" top="81" bottom="224">
      <gBSDUnit xsi:type="gbsdsa:gBSDSliceUnitType"
        syntacticalLabel=":I-Slice:0"
        start="576" length="7802" marker="disposable"
        left="0" right="352" top="0" bottom="80" />
      <!-- ... and so on... -->
    </gBSDUnit>
    <!-- ... and so on... -->
  </dia:Description>
</dia:DIA>
```

In the specification, DIA tools are clustered into several categories. For example, Bitstream Syntax Description (BSD) describes the syntax of a binary media resource such as video. With Bitstream Syntax Description Language (BSDL), which is an XML Schema based language and standardized in the MPEG-21 framework, it is possible to design specific Bitstream Syntax Schema (BS Schema) describing the syntax of a particular coding format. A normative

processor named BSDtoBin is specified in MPEG-21 to generate the adapted bitstream for BSDL. Since BSDL provides a way for describing bitstream syntax with a codec specific BS Schema, an adaptation engine consequently requires knowing the specific schema. Therefore, a generic Bitstream Syntax Schema (gBS Schema) is specified in the MPEG-21 framework to offer a format independent adaptation procedure. The gBS Schema introduces the means to describe hierarchies of syntactical units and addressing means for efficient bitstream access. A description conforming to this schema is called a generic Bitstream Syntax Description (gBSD), which provides an abstract view on the structure of the bitstream that can be used in particular when the availability of a specific BS Schema is not ensured. The gBSD is essentially a metadata definition of the media, written in the form of XML. For example, as seen in Box 1, the gBSD represents each frame's number, starting position in the bitstream, frame type, and length of each frame among other information for a video bitstream.

Benefits of applying gBS Schema and using gBSD are manifold, which includes, but is not limited to, the following – It enables coding format independence, and the description represents arbitrary bitstream segments and parameters, which allows random access into a bitstream. The bitstream segments may be grouped in a hierarchical way allowing for an efficient hierarchical adaptation.

2.3. H.264 Video

In the past decade, a fair number of video codecs like H.261, MPEG-2, and H.263 have evolved. H.264 is the latest video coding and compression standard by ITU-T and ISO/IEC. The advanced compression technique, improved perceptual quality, network friendliness and versatility of the codec [15][16], drives it to outperform all the previous video coding standards. Instead of detailing all the features of H.264 video, the features that have been used to achieve the objective are abstracted in the following two paragraphs.

One of the major reasons behind choosing H.264 is its multiple reference pictures for motion compensation. Motion compensation is a way of describing the difference between

consecutive frames in terms of where each section (e.g. macroblock) of the former frame has moved. H.264 coding standard extends the enhanced reference picture selection technique of H.263++ (a.k.a. H.263v3 or H.263 2000) which allows selecting among a larger number of pictures that have already been decoded.

In H.264, slice sizes are flexible so that a picture can be split into one or several slices. Slices are self-contained and can be decoded without using data from other slices. Every slice consists of a group of compressed macroblocks and begins with a slice header. In the H.264 specification, there is a special feature called Flexible Macroblock Ordering (FMO). Within the FMO, there can be at most eight slice groups, which eventually restrict the available cropping options for spatial adaptation. In addition, many widely used decoders do not implement some of the advanced features of H.264, like FMO. Therefore, we devised our own slicing strategies for spatial adaptation.

H.264 offers an entropy coding design that includes Context-Adaptive Binary Arithmetic Coding (CABAC) and Context Adaptive Variable Length Coding (CAVLC). In order to achieve byte-alignment, bit sequences are being padded by the encoder whenever necessary. In this thesis, the sliced architecture and the multiple frame dependency features are exploited for compressed-domain spatial and temporal adaptation. For compressed domain operations, we analyzed how H.264 classifies the bits that comprise a compressed video sequence.

2.4. Integer Linear Programming

In this thesis, we followed a heuristic approach guided by some predefined rules in order to generate a P2P streaming overlay. However, in general, a P2P streaming system can be seen as a constraint system, and maximizing available overlay resources refers to an optimization problem. Therefore, a mathematical model of the adaptive streaming system is also formulated, where the efficient streaming is defined as an optimization of a cost function that can be solved as a Linear Programming (LP) problem.

In an Integer Linear Programming (ILP) problem, a given linear function is maximized or minimized on the basis of a collection of linear inequalities on a number of integer variables. A binary (0-1) Integer Linear Programming (a.k.a. Pseudo-Boolean) problem is a special case of ILP in which all variables are restricted to take values of either 0 or 1. In other words, a linear 0-1 term can be maximized or minimized with respect to a set of linear constraints over 0-1 variables. In general, Pseudo-Boolean linear integer-programming problem can be written as follows:

Maximize:

The objective function, CL

Here, C is the vector of coefficients and L is the vector of variables

Subject to:

A set of constraints, S

Here S is defined as: $A.L \leq b$

where, A is a matrix of coefficients, and for binary ILP, $L_i \in \{0, 1\}^n$

Example:

Maximize: $c_1l_1 + c_2l_2 + c_3l_3$

Subject to: $a_{11}l_1 + a_{12}l_2 + a_{13}l_3 \leq b_1$

The basic idea of solving Pseudo-Boolean constraints with 0-1 Integer Linear Programming is to solve first an integer relaxation of the original problem, where variables are allowed to take more values instead of being restricted only to Boolean values. A pure cutting plane algorithm can be used to reformulate the problem until the relaxation of the problem is sufficiently strong and solves the original problem [17]. This procedure therefore implies reformulating a set of constraints until 0-1 satisfiability becomes easily decidable. As an example,

given a polytope and a real-valued function $f(x_1, x_2, \dots, x_n)$ defined on this polytope, the goal is to find a point on the polytope where this function has the largest or smallest value. In the above case, replacing the condition $L_i \in \{0, 1\}^n$ by $L_i \in [0, 1]^n$ is a linear programming relaxation where L can be placed on the polytope. In literature, several methods have been used to find a solution based on the relaxation and satisfying the 0-1 constraint, e.g. Branch and bound [18], Branch and cut [19], Generalized Davis Putnam (GDP) [20], Simplex method [21], etc. Each of these methods is useful for one specific class of problems. For example, GDP method is suitable for problems with a fewer number of variables and more constraints.

2.5. Video Watermarking and Perceptual Encryption

Video watermarking methods can be classified into spatial-domain, transform-domain, and compressed-domain approaches. In spatial-domain watermarking, the watermark is embedded directly in the pixel domain. Techniques involving spatial-domain watermarking are theoretically straightforward and demonstrate low time complexity during watermark embedding and detection steps. On the downside, these techniques fail to meet adequate robustness and imperceptibility requirements. In transform-domain watermarking, the host signal is transformed into a different (e.g. frequency) domain first, and then the watermark is embedded in selective coefficients. Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT) are two examples of widely used transformation techniques. Watermark embedding in the transform domain is advantageous in terms of visibility and security. In compressed-domain watermarking approaches, the watermark bits are either directly inserted into the encoded bitstream, or in a partially decoded bitstream. Benefits are that these approaches are computationally less expensive and require less processing power.

Encryption is a process of scrambling or converting data known as Plaintext into a form, called a Ciphertext that cannot be interpreted by an unintended user or receiver. The simplest way to encrypt a video is perhaps to consider the whole stream as a 1-D array and then encrypt this 1-D stream with the encryption key. However, with the advancement of time, video

encryption mechanisms considering sensitivity of digital video have been developed. The notion of perceptual encryption refers to partially degrading the visual data of the video content by encryption. The term perceptual encryption is used for the adaptation framework in this thesis because after encrypting selective slices based on the region of interest, the perceptual quality of the content is degraded and the encryption goal is achieved.

CHAPTER 3

CURRENT STATE OF KNOWLEDGE AND RELATED WORK

There have been many research activities and advances in video adaptation, video authentication, video encryption, and overlay generation in the past decade. Prominent research works identified during the background study are highlighted in this chapter.

3.1. Video Adaptation

Video adaptation is a promising meadow for challenging pervasive media applications. Earlier works, for instance [22] and [23], explored some interesting aspects of adaptation such as bandwidth reduction, format conversion, and modality replacement for Web browsing applications. In [22], authors comprehended that the client devices with limited communication, processing, storage and display capabilities cannot easily handle much of the rich multimedia content. In order to improve content access, authors outlined a system for scalable delivery of multimedia which uses an InfoPyramid for managing and manipulating multimedia content composed of video, images, audio, and text. Recently, international standards such as MPEG-7 [24], W3C [25], and TV-Anytime [26], have developed related tools and protocols to support development and deployment of video adaptation applications. Different standards are targeted for different applications. For example, TV-Anytime focuses on adaptation of content consumption into high-volume digital storage in consumer platforms such as personal video recorders. W3C and IETF focus on facilitating server/proxy to make decisions on content adaptation and delivery.

In terms of architecture, techniques, quality optimization, complexity reduction, and watermark insertion, a comprehensive overview of digital video transcoding is presented in [27]. In [28], authors investigated quality adaptation algorithms for scalable encoded variable bit rate

video over the Internet. The goal was to develop a quality adaptation scheme that maximizes perceptual video quality by minimizing quality variation, while at the same time increasing the usage of available bandwidth.

Cascaded transcoding schemes for downscaling videos are quite established. Until very recently, most of the video adaptation schemes applied cascaded transcoding operations for temporal and spatial adaptation [31]-[34]. As time progressed, these schemes happened to be suitable only for off-line applications. Cascaded schemes provide high quality output video but at the cost of high transcoding time; for example, Zhang et al. proposed a method in [32] where first, the video is decoded and then it is downscaled in the pixel domain. During re-encoding, a mode-mapping method is employed to speed up the process where mode information of the pre-encoded macroblocks is used to estimate the mode of the corresponding macroblock in the transcoded bitstream. Cock et al. proposed a similar downscaling scheme in [33] that uses transcoding. In addition to applying mode-mapping strategy, this scheme reduces re-encoding complexity by performing motion compensation only in the reduced resolution domain.

Bonuccelli et al. [29] introduces buffer-based strategies for temporal video transcoding, adding a fixed transmission delay for buffer occupancy in frame skipping. A frame is skipped if the buffer occupancy is greater than some upper value, and it is always transcoded if the buffer occupancy is lower than some lower value provided the first frame (i.e. the I-frame) is always transcoded. The Group of Pictures (GOP) level rate adaptation scheme is introduced in [30] for a single stream, variable target bitrate H.264 encoder. The scheme in [30], allows each group of pictures to be encoded at a specified bitrate using a dynamically updated table to select the starting quantization parameter for each GOP. Block Adaptive Motion Vector Resampling (BAMVR) method is proposed in [31] to estimate motion vector for frame rate reduction in H.264. However, the transcoder follows straightforward cascading architecture of the decoder and encoder. Devillers et al. propose a BSD based adaptation in streaming and constrained environments [36]. In their framework, the authors emphasized on BSD based adaptation applying BS Schema and BSDtoBin processors. In this research, gBSD has been used, which

provides an abstract view on the structure of the bitstream that can be used in particular when the availability of a specific BS Schema is not ensured. In [37], Deshpande proposed adaptive frame scheduling for video streaming using a fixed frame drop set. Sender adjusts the deadline of an important frame, which is estimated to miss its deadline by dropping less important next frame(s), and sends the deadline-adjusted/postponed frame to be displayed in place of the next dropped frame(s). However, the visual quality of the reconstructed video stream on the receiver side may not be acceptable for those videos having high motion or frequent scene change. To overcome this issue, the technique described in this thesis uses individual frame importance. In addition, frames are managed in groups called Frameset so that after transmission and adaptation, every frameset is self-contained.

Wang et al. [34] and Arachchi et al. [35] have discussed spatial adaptation techniques by cropping. In [34], authors employ a set of transcoding techniques based on the H.264 standard to crop the region of interest in each frame. In this process, compressed H.264 bitstream is first decoded to extract the region of interest. The cropped video stream is then re-encoded. The region of interest is determined by using an attention-based modeling method. As we can see, the above-mentioned scheme applies cascaded operations for spatial adaptation. For obvious reasons, this scheme is suitable for offline applications only. The spatial adaptation approach portrayed in this thesis applies the cropping concept but without any cascaded operations. In [35], authors use a similar transcoding process for cropping H.264 videos. Additionally, this scheme reduces transcoding complexity by using a special process for encoding SKIP mode macroblocks in the original video. When re-encoding these macroblocks in the cropped video, the transcoder compares the motion vector predictors (MVPs) for the macroblock in the original video with the computed MVPs in the cropped video. If the MVPs are the same, then SKIP mode is selected for the macroblock. Thus, the transcoding complexity is reduced by avoiding the expensive rate–distortion optimization process to detect the macroblock modes for SKIP mode macroblocks in the input video.

Online and real-time video adaptation is a means to enable the UMA concept of ubiquitous access to multimedia contents. Recently, researchers have investigated multiple-proxy based transcoding solutions (e.g. [38]) to scale a video for heterogeneous environments. However, to save time and cost, it is practical to perform adaptation operations in one adaptation service point rather than forwarding data packets to different proxies. In this thesis, the proposed adaptation scheme avoids dedicated adaptation servers/proxies; rather uses the off-the-shelf personal computers for live adaptation operations.

Most recently, scalable video coding (SVC) is in the center of interest to achieve adaptability of the coded video sequence where the compressed video is coded into layers – the base layer is coded at low quality and then one or more enhancement layers are coded to gain high quality. Thereby, the adaptability of the coded sequence is obtained by changing the number of enhancement layers transmitted from the sender side. In the literature, there are several papers summarizing the above concept (e.g., [39][40]), and extending it for different scenarios such as in-network adaptation with encryption [41] and adaptation of the SVC stream on a network device [42]. However, concern with the SVC is that it can only achieve bitrates in a limited set usually decided at coding time.

3.2. Video Watermarking and Encryption

Vetro and Timmerer revealed several emerging research topics and open issues related to digital content adaptation including permissible and secure adaptation, as well as emphasized on secure adaptation of encrypted contents in [13]. Compressed-domain video watermarking and encryption along with adaptation is an attempt to address few of these important issues.

For video authentication, watermarks can be embedded in the uncompressed domain during or after the compression process. The DCT coefficient based embedding systems [43]-[45] embed binary watermark bits in the DCT domain derived from different extracted features, for example, in [43], a human visual model is adapted for a 4×4 DCT block, and in [44], relations between predicted DCT coefficients and real DCT coefficients are considered to embed the

watermarks. A watermarking method proposed by Qiu et al. in [45] embeds a robust watermark into the DCT domain and a fragile watermark into motion vectors during H.264 compression. Zhang and Ho in [46] propose a scheme that uses the tree-structured motion compensation, motion estimation, and Lagrangian optimization of the H.264 standard. The authentication information is represented by a binary watermark sequence and embedded into video frames. Dima Pröfrock et al. [47] propose a new transcoder, which analyses the original H.264 bit stream, computes a watermark, embeds the watermark for hard authentication, and finally, generates a new H.264 bitstream. All the above techniques either embed watermarks during the encoding process of the H.264 video [45][46] or employ cascaded decompression and recompression operations [43][44][47] to analyze H.264 bitstream and embed the watermark. In this research, the watermark will be embedded in compressed domain without any cascaded operation.

Extensive research has been performed [48]-[50] for video encryption based on perceptual encryption. Researchers have proposed scalability-based perceptual encryption, perceptual encryption for wavelet-compressed images and videos, perceptual encryption of motion vectors in MPEG-videos, etc. Partial or selective video encryption algorithms are also investigated which are usually applied in selective layers, e.g. [51]-[53]. In partial or selective video encryption algorithms, the content is encrypted and then decrypted by exploiting the inherent properties of the video coding layer(s). However, none of these proposals, i.e. [48]-[53], is compatible with format agnostic compressed-domain video processing in an intermediary node. A comprehensive and promising work for H.264/AVC video encryption is reported in [48], where authors use two algorithms – one for encrypting the headers, and a second for encrypting the slice payload. The objective of the authors in [48] is similar to the encryption scheme presented in this thesis; however, the only difference here is that the proposed encryption technique will survive compressed-domain adaptation operations in any intermediary node. In [54], authors detailed a compressed-domain temporal adaptation resilient macroblock-based encryption. Compared to [54], the encryption scheme presented here encrypts slices, and survives both temporal and spatial adaptation operations.

3.3. P2P Streaming

P2P architectures have received a great deal of attention in both academia and industry [55]-[60] due to their scalability and alleviation of the need for powerful centralized resource servers. However, such a promising concept comes with a new family of challenges. Several such issues pertaining to the evolutionary dynamics (e.g. resource usage, stochastic optimization, etc.) have already been addressed by the researchers (e.g. [61][62]). Interested readers are referred to [6] for a detailed review of the P2P streaming and existing application layer multicast (ALM) protocols. In terms of content distribution, the most common approach for P2P streaming is to construct one or several multicast trees to distribute the stream between the source and different users (e.g. [63][64]). Another approach lets the peers self-organize in a mesh and request different portions of the video from their neighbors, with no particular emphasis on the structure of the distribution path (e.g. [65]). In a traditional decentralized design, peers communicate directly with each other for sharing and exchange of data as well as other resources such as bandwidth. A problem with the decentralized solution is that a node may overwhelm other neighboring nodes (e.g. AMMO [66]). Exarchakos and Antonopoulos propose an unstructured P2P overlay to share the network resources in [67]. Their design, however, considers dedicated servers and utilizes peers to reduce server load, whereas the goal in this thesis is to avoid dedicated servers and utilize the idle resources of the participating peers.

Recent studies on real-world P2P streaming systems like [55] and [60] have demonstrated that the streaming performance can be typically maintained at a high level once the overlay has reached a reasonable scale in an optimum setting; however, this is challenged by peer contribution. In [55], authors do not consider application level multicast and assume stored videos where the video traffic is a known priori. Compared to [55], the adaptive streaming design presented in this thesis is based on ALM, and user environment characteristics are not a known priori. Error resilient transport for P2P video streaming is covered in [70]. In [70], authors focus on broadcast categories of robust video streaming schemes such as forward error correction, prioritized automatic repeat request, etc. Tan et al's work [71] focuses on network delays between

source and receiver while building multicast trees for multimedia streaming. However, if we consider the latency added during video capture, encoding and adaptation, the network delay is very small and less important than adaptation requirements or resource limitations.

3.4. Video Streaming

To meet the demand of P2P video streaming, many first generation applications have appeared on the Internet, such as PPLive, PPStream³, etc., claiming tens of thousands of simultaneous users in a single channel, with stream rates between 300 kbps to 1 Mbps. However, their protocol design and encoding structures have not been targeted at peer heterogeneity. Moreover, their unstable video quality does not offer a high-quality viewing experience.

Layered encoded video streaming has received a significant research interest in the multimedia community in the last couple of years. PALS [75], for example, is a receiver-driven approach for quality adaptive playback of layer encoded streaming media from a group of congestion controlled sender peers to a single receiver peer. Since in layered encoding only a limited number of layers can be stored for each video, end nodes might get less quality than what they can handle.

In ZIGZAG [8], authors propose a video streaming strategy where the peers are clustered into a hierarchy with the new concept of head and associate head for each cluster. The work by Xiaofeng et al. [76] is one of the few video streaming systems that uses ordinary computers (peers) as servers rather than using a single or a few dedicated servers. In [76], each video is first encoded into multiple descriptions and then each description is placed on a different server. The solution is acceptable if there is some collaboration between peers before the streaming starts; however, it is inconvenient for an isolated peer to initiate the streaming session and to place the video descriptions in different participating nodes.

³ www.pplive.com , www.ppstream.com

In the literature, there are also prominent research works entailing live media streaming (e.g. [68], [69]). In [68], authors explore and chart large-scale P2P live streaming sessions, and their topological characteristics. In [69], authors focus on how to construct a stable multicast tree to minimize the negative impact of frequent node departures in an overlay as well as how to recover from errors due to node/network failures. Some of the existing research works entailing P2P streaming (e.g. [72][73]) emphasize multiple senders because of the limited capacity, unreliability of peers, and also to achieve disjoint network path streaming (e.g. [74]). Similarly, in this thesis, the frequent node departure problem in live P2P streaming sessions has been taken care of by applying a multiple parent approach and dividing the live stream into small clips.

3.5. Adaptive Video Streaming

From the experience of developing an adaptation system and from [70], it has been observed that significant processing gains can be obtained when video streaming systems are capable of adapting the content based on the encoding structure. More details on the necessity and benefits of adaptation in mobile environments are discussed in [77]. In the literature, there are prominent architectures for P2P streaming solutions but none of them completely overcomes the challenges from the dynamic P2P environment. Some of these challenges arise due to peer heterogeneity and peer mobility. To solve these challenges, simultaneous adaptation and streaming of video contents in a distributed manner is therefore a new concept. In the literature, there are not many promising works in this topic except the PAT system [78], proposed by Liu et al. In the PAT system, authors suggest saving the intermediary transcoding results as meta-data and re-use the metadata for a later transcoding request. The incentive behind PAT and this research proposal are similar. Nevertheless, the PAT scheme is substantially different in a number of aspects: i) The system to be presented in this thesis will not need meta-data overlay because transformation effort of the gBSD to adapted gBSD is very small compared to that of bitstream. ii) The new system will provide a complete solution to adaptation and streaming utilizing MPEG-21 gBSD for H.264 video. gBSD provides the flexibility to adapt video contents even outside the streaming system. iii) The presence of backup parents is not needed in the

proposed design because the tree-controller re-assigns a parent when the current parent leaves the system. In PAT, authors report that meta-data takes up to 6% additional bandwidth and that the metadata size is 20-25% of the original video file size for frame rate transcoding. In contrast, gBSD takes up only 2-5% of the compressed bitstream and gives details of the bitstream syntax structure.

Impact of the characteristics of the streaming source and participating peers to the perceived video quality and the playback delay has been highlighted in [79], where authors focus on the impact of the streaming rate, source upload bandwidth, and bandwidth heterogeneity on transmission and playback delay. The performance analysis in Chapter-9 shows that the presence of powerful nodes with higher contribution resources in the system plays a significant role for the success of an overlay.

The idea of built-in incentives in live video dissemination is not new, rather it was introduced in early research works involving live P2P video streaming (e.g. [63][80]) in order to overcome the available resource related issues. Several studies have shown that the tit-for-tat approach is not sufficient to prevent free riders or fully incentivize peer (e.g. [81][82]). Tit-for-tat creates a differentiated service at the application layer, providing high-speed uploaders with short download times and low-speed uploaders with high download times. In this thesis, a minimum contribution rate method is devised where the tit-for-tat concept is avoided, and a peer's actual capability and global knowledge of similar peers has been taken into consideration.

Given a video source, a group of intended destination peers and heterogeneous network resources, the challenge is to distribute the content among these peers to ensure the best quality of experience using P2P paradigm. From the above discussion, it is prevalent that there exists promising solutions to address this issue; however, without an optimal solution it is hard to judge the efficiency of the existing protocols. Since the dynamic traffic of the live media stream cannot be predicted accurately, one can resort to stochastic models (e.g. [83]) for robust network optimization. In [11], authors describe a stochastic model to compare different data-driven downloading strategies for P2P streaming. The model considers two performance metrics,

probability of continuous playback and startup latency, to evaluate chunk selection strategies in P2P streaming. In this model, authors assume independent and homogeneous peers in a symmetric network setting. Compared to [11], the objective of the proposed analytical approach is to model multi-parent video chunk dissemination in a P2P setting, together with peer heterogeneity and network asymmetry in order to emulate the real-world scenario. In the literature, combinatorial optimization problem has been considered (e.g. [84][85]) as a way of finding the optimum solution towards multicast tree generation problem. Compared to [84] and [85], the approach presented in this thesis is different in terms of both modeling and optimization goal aspects.

CHAPTER 4

COMPRESSED-DOMAIN VIDEO ADAPTATION, AUTHENTICATION, AND ENCRYPTION

A systematic procedure for designing video adaptation framework involves identifying the adaptation entities (e.g. pixel, frame, group of pictures, etc.) for adaptation as well as identifying a feasible adaptation technique (e.g. re-quantization, frame dropping, etc.) [86]. We also need to develop a method to estimate the resource and the utility values associated with the video entities undergoing adaptation. In this thesis, the goal of compressed-domain adaptation is to adapt video frames for a target screen size and/or frame rate. To meet the user preference or device requirements, a video is thus scaled down by reducing frame rate and/or spatial size. An adapted version with a lower frame rate is retrieved by removing segments belonging to the temporal dimension. Similarly, a version with a lower spatial resolution is retrieved by removing segments that belong to the spatial dimension. For temporal adaptation, I-frames and P-frames are the entities; and skipping frames from the original compressed bitstream is the chosen technique. Similarly, for spatial adaptation, selective slices from a region of interest are dropped to acquire the desired spatial resolution. The design decision behind choosing these techniques is made with the goal of creating a simple and computationally efficient adaptation module.

4.1. Compressed-domain Video Adaptation

Compressed-domain video adaptation procedures can be divided into two major parts as shown in Figure 5 – Part-1: Compressed video and metadata generation which is performed during the encoding phase; Part-2: Adaptation of the metadata and compressed video which is performed in an intermediary node. Tasks involved in Part-2 can be further divided into two sub-processes: i) Metadata transformation, and ii) Adapted video generation based on the transformed metadata. Video adaptation takes place on any intermediary nodes such as peers,

gateways, and proxies; however, server-based solutions will not scale beyond a certain population of receivers since the servers may become the service bottleneck.

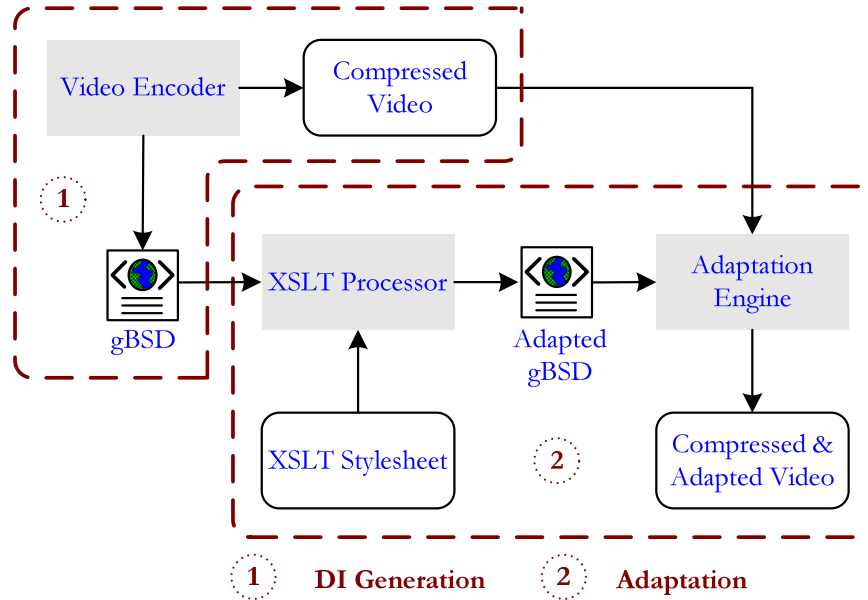


FIGURE 5. GBSD-BASED ADAPTATION ARCHITECTURE

Practically, the content metadata in the form of MPEG-21 gBSD is the representation of the actual video bitstream in the form of an XML document. gBSD does not replace the actual bitstream but provides additional information regarding the bit/byte positions of syntactical and semantic levels of a video. Eventually, gBSD does not necessarily provide any information on the actual coding format; however, codec specific information enables codec-agnostic adaptation in an intermediary node. While gBSD describes the high-level structure of the bitstream, similarly, the Usage Environment Description (UED) represents information about the usage context, terminal capabilities (e.g. codec capabilities), network conditions (e.g. maximum or minimum bandwidth), user characteristics (e.g. personalized requirements) etc. A sample UED document is shown in BOX 2.

BOX 2. UED: DEVICE POWER CHARACTERISTICS

```
<?xml version="1.0" encoding="UTF-8" ?>
<dia:DIA
  xmlns:dia="urn:mpeg:mpeg-21:2003:01-DIA-NS"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <dia:Description xsi:type="dia:UsageEnvironmentType">
    <dia:UsageEnvironmentProperty xsi:type="dia:TerminalsType">
      <dia:Terminal>
        <dia:TerminalCapability xsi:type="dia:PowerCharacteristicsType"
          batteryTimeRemaining="0" batteryCapacityRemaining="0"
          runningOnBatteries="true"/>
      </dia:Terminal>
    </dia:UsageEnvironmentProperty>
  </dia:Description>
</dia:DIA>
```

In the gBSD, the hierarchical structure of the encoded video (e.g. frame numbers along with their starting byte number, length of the frame, and frame type) is included. For each frame, the slice data information is also included in the gBSD. Related information of a slice includes the starting byte number of the slice, the length of the slice in bytes, and the region of the video frame where the slice belongs. A marker “essential” or “disposable” is added for each slice as can be seen in Box 1 (in Section 2.2).

The Digital Item (i.e. video bitstream along with its gBSD) is the basic content for the content provider on the delivery path. In the MPEG-21 framework, generation of the original Bitstream Syntax Description from binary data (a.k.a. BintogBSD) is not normatively specified. Therefore, gBSD is generated during the encoding process of the bitstream because of the fact that the structure of the encoded bitstream can be well preserved while encoding. To do this, the ITU-T reference software implementation JM 9.5 [87] has been enhanced with gBSD generation

functionality. In addition, the method for the transformation of the gBSD is not specified in the DIA standard. Therefore, Extensible Stylesheet Language Transformations (XSLT) [88] has been utilized. XSLT is a declarative, template based, transformation language for XML documents. An XSLT processor is placed in the adaptation module to transform the original metadata. An XSLT processor needs two inputs: i) An XSLT style sheet that contains the template rules and describes how to display a resulting document, ii) The input XML document represented as DOM⁴ (Document Object Model) tree. In addition, a set of parameters and parameter values can be passed to the XSLT processor to steer the transformation. Once the adaptation operation is initiated, the XSLT processor traverses the DOM tree and applies the changes according to the transformation rules defined in the style sheet.

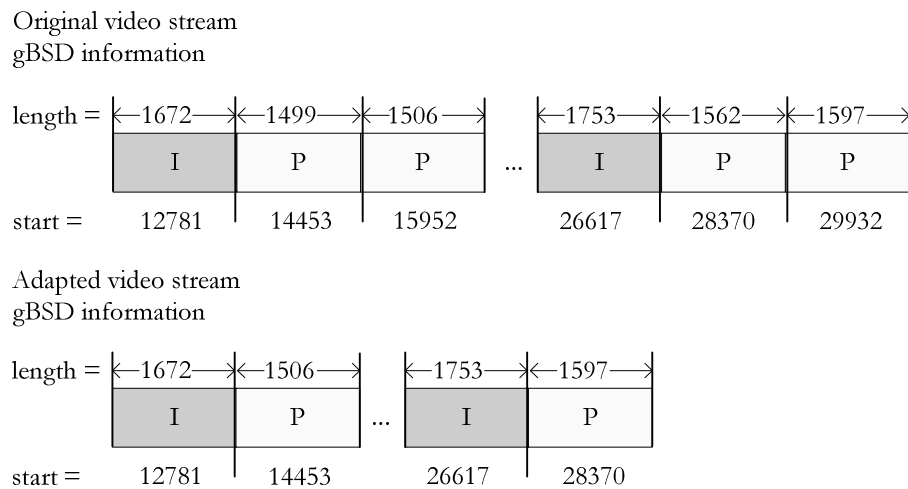


FIGURE 6. GBSD TRANSFORMATION FOR TEMPORAL ADAPTATION

4.1.1. Temporal Adaptation

For temporal adaptation, a frame skip pattern is implemented in the XSLT template based on the original frame rate and required frame rate, which is matched against the nodes in

⁴ <http://www.w3.org/DOM/>

the source tree of the gBSD. In Figure 6, an example of the gBSD transformation for temporal adaptation is illustrated. The frame skip mechanism enables consistent frame dropping pattern for a target frame rate in between 1 to 29fps if the original encoding frame rate is 30fps. For example, for any Digital Item, if the target frame rate is 10fps then the selected frame numbers to be dropped will be same for all DIs. The pseudo-code of this strategy is shown in Box 3. For the frame skip, in general, it is assumed that the new rate would be less than the old frame rate and greater than zero. If the new rate is greater than or equal to the old frame rate, then the source tree becomes the resulting tree and copies all nodes. Finally, a bitstream parser parses the adapted gBSD and then discards those parts from the original compressed bitstream that are missing in the transformed gBSD followed by updating necessary header information for format compliance.

BOX 3. PSEUDO CODE FOR FRAME DROPPING

```
FrameSkip(newFrameRate,oldFrameRate)
{
  a=1, b=1;
  temp = newFrameRate/oldFrameRate;
  if(newFrameRate<oldFrameRate && newFrameRate!=0)
  while(a<oldFrameRate)
  {
    copyFrame(a);
    a = ceiling(1+b*temp);
    b++;
  }
}
```

4.1.2. Spatial Adaptation

For spatial adaptation, we devised our own slicing strategies. For spatial adaptation, it is preferable to have a smaller number of slices for each frame. The reason is that if we increase the

number of slices, then it will eventually increase the overhead due to the added number of slice-headers. On the plus side, using a large number of slices in each frame gives more region-based cropping options to the adaptation module. To facilitate spatial adaptation by means of dropping slices, we have devised three slicing strategies to be applied during the initial DI generation: i) Block-Strategy, ii) Wide-Strategy, and iii) O-Strategy, as shown in Figure 7. The Block-Strategy and the Wide-Strategy is appropriate for a video with a large area of interest, whereas the O-Strategy may suffice for a video with a smaller area of interest (e.g. talking head). In Figure 7, shaded slices represent the disposable slices, which may be dropped to achieve the target resolution, and rest of the slices (i.e. white areas) represents the essential slices referring to a region of interest.

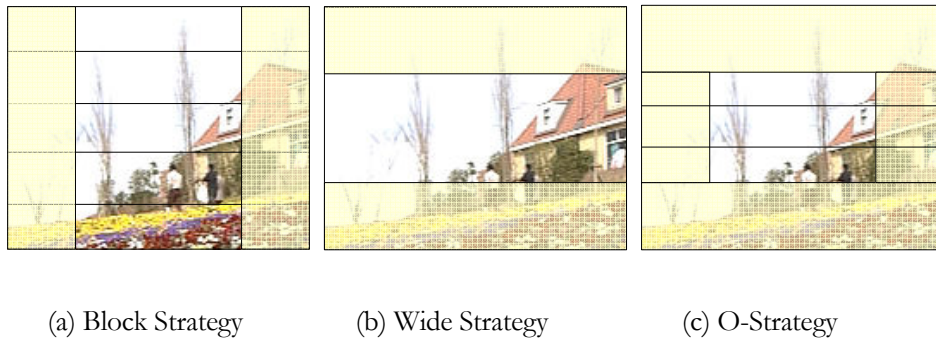


FIGURE 7. SLICING STRATEGIES

Within the coded bitstream, only the slice(s) belonging to the desired region of interest is served to the client. Similar to the temporal adaptation, a bitstream parser in the resource adaptation module detects the disposable slices in the video frames by parsing the adapted gBSD and then removes those slices from the compressed bitstream. The Sequence Parameter Set (SPS) of the video stream needs to be updated before sending the adapted bitstream to the client in order to indicate that the video file has been adapted so that the video players can display only the desired region accordingly. For spatiotemporal adaptation, the final output from the adaptation module is a H.264 format compliant adapted compressed bitstream and its corresponding adapted metadata. Therefore, any standard H.264 player can decode and play the

videos without any knowledge of the adaptation scheme. The adapted metadata may be discarded right after adaptation completion unless there is need of further adaptations of the adapted bitstream.

4.1.3. gBSD-based Adaptation of Continuous Streams

For XSLT transformation of gBSD, complete XML description of a video file needs to be loaded before being adapted. Now, generating the metadata description of a long video file is inconvenient for transmission to an intermediary node for future adaptation operations. As a result, to apply the above-mentioned compressed domain adaptation scheme for continuous streams, video streams need to be processed as small segments/clips. Metadata description for each of these segments/clips also needs to be generated during the DI generation process. Benefits of this approach are as follows:

- For variable bit rate in the transmission channel, modified bit rate requirements can be applied to the next available video segments.
- The intermediary nodes do not need to save the video segments for a long time and may discard right after re-transmitting and/or adapting to other user(s).
- If the network characteristics or user preference changes during a streaming session, a new parent can be assigned to serve that user.

For live video feeds from a camera, clips may need to go through type conversion and pre-processing before putting it in the processing module to compress, adapt and transmit. Therefore, a live streaming request may require a fixed delay for pre-processing and adaptation of the clips. A sequence diagram for live streaming is shown in Figure 8.

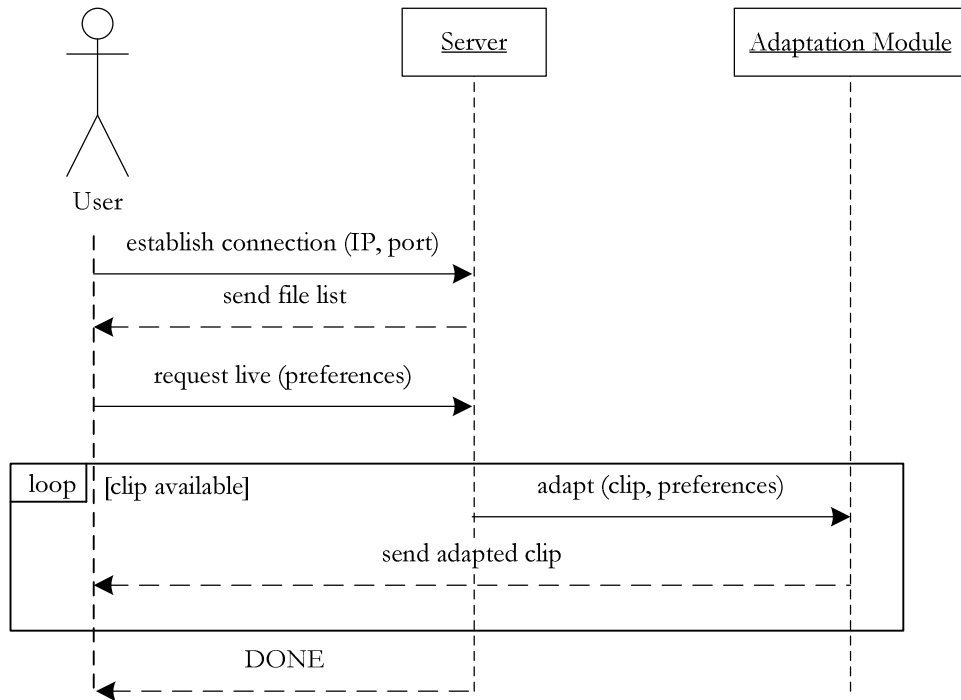


FIGURE 8. SEQUENCE DIAGRAM OF A LIVE VIDEO REQUEST

4.2. Compressed-domain Video Authentication and Encryption

In the wake of a technological challenge to prevent piracy, the entertainment industry has started using Digital Rights Management (DRM) technologies to protect contents. Consequently, these content security technologies limit the content adaptation possibilities in an intermediary node. Let us say for transmission of sensitive video contents or to ensure revenue for a copyrighted item, video contents can only be adapted at trusted adaptation servers since they have to be fully decrypted before performing the necessary adaptation operations. As a result, it is not convenient to adapt such encrypted contents on the fly. On the other hand, small handheld devices with multimedia content processing capabilities are being used for live streaming and other day-to-day media-based services like video surveillance. Keeping in mind the

sensitivity of some of these contents, encryption and/or authentication need to be applied to ascertain trustworthiness. In this section, a metadata-assisted authentication and encryption technique is presented. To authenticate live video, digital signature is embedded in real time to scrutinize the integrity of the received content in a contained environment, whereas the encryption system can be merged to hide/secure information (e.g. in video surveillance systems) or to reduce perceptibility of a video content (e.g. free preview of a pay-per-view movie). Such schemes are beneficial for an end-to-end video delivery to heterogeneous environments in any video distribution system.

4.2.1. Video Authentication

The authentication scheme proposed in this thesis embeds the authentication bits in the bitstream after the adaptation and in the compressed domain. To embed watermark in live video streams, it is convenient to compute and insert the digital signature for the live video stream in small parts. This approach saves computation time and effort for the signature computation of individual frames. This particular technique is useful in video surveillance systems where video data being captured by wireless/dispersed cameras is transmitted to distant receivers in a heterogeneous network. Towards this, an authentication technique is devised, which uses a fragile watermarking scheme. The technique uses gBSD to select the marking space directly from the compressed H.264 video rather than decoding any part of the media. The watermarking procedure presented here is synchronized with the adaptation operations so that both can be done simultaneously.

In Section 4.1.3, we have learned that to offer metadata-based adaptation, live video streams are processed as small clips. In the video recorder, the encoder encodes the raw video input to H.264 video and generates the gBSD for each clip as well. Therefore, a watermark embedder added to the adaptation engine embeds authentication bits during the adaptation procedure. The benefits of this approach includes – i) The gBSD is parsed only once while adapting, ii) Authentication bits are computed and embedded in the adapted frame(s). Figure 9 summarizes the video capture, DI generation, adaptation, and watermark embedding.

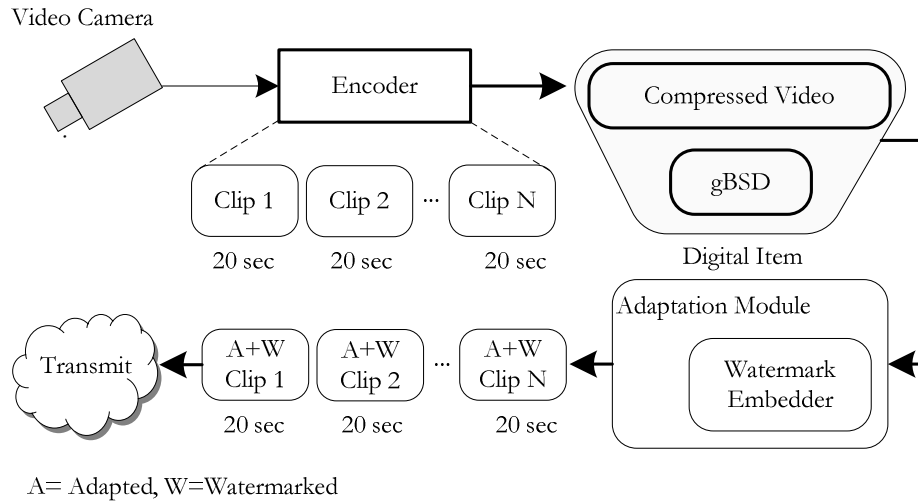


FIGURE 9. PROPOSED AUTHENTICATION ARCHITECTURE

a. Watermark Embedder

The watermark-embedding module is implanted inside the adaptation engine to embed authentication bits on the fly while adapting. In this regard, a 5-step process is followed and is described below:

Select the marking space. From the gBSD, marking space can be selected from available alternatives such as frame, slice, macroblock, and block (Figure 10). Application specific marking space can be selected in a predefined way and a fixed watermark embedder can be designed. Otherwise, if marking space is selected manually, the watermark embedder should be capable of inserting watermark bits in the selected segment directly in the compressed bitstream. For manual selection of marking space, start and length of each segment need to be defined in the gBSD. Finally, selection of a marking space and applying customized modification must conform to H.264 bitstream specification to avoid incompliance for a standard player or decoder. In the system presented here, slice data of frames from the individual video clips are used to compute the authentication bits and finally these bits are embedded in the slice header.

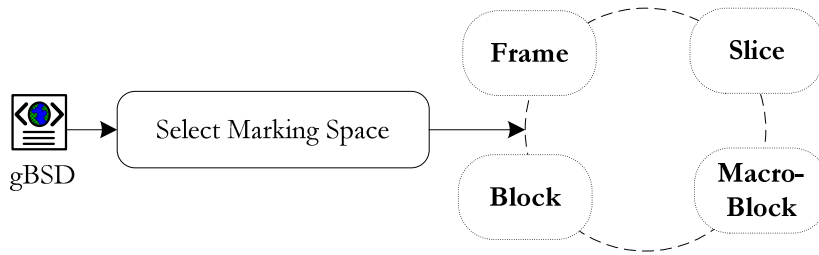


FIGURE 10. MARKING SPACE SELECTION FROM GBSD

Calculate the size of the marking space. Marking space size calculation helps to figure out the length of the hash value to be computed from each compressed video clip in advance. In H.264 video coding, a picture can be split into one or several slices where slice sizes are flexible. Slices are self contained and can be decoded without using data from other slices. Slice header VLC byte-align bits (minimum 1 bit and maximum 7 bits) can be utilized to embed the authentication bits. But the efficiency depends on the video length, frame rate, and the total number of byte-align bits in a clip. So the size of the marking space (M_{space}) can be calculated as follows:

$$M_{space} = \text{VideoLength} \times \text{FrameRate} \times \text{VLCByteAlignBits}$$

Compute the Hash value. A hash value can be computed based on the individual frame data or for the whole video clip. Essentially the size of hash value depends on the size of the marking space. A private key is an essential input to the hash function (Figure 11.A). Importantly, complicated or robust hash functions will require higher execution time in order to compute and embed the authentication bits.

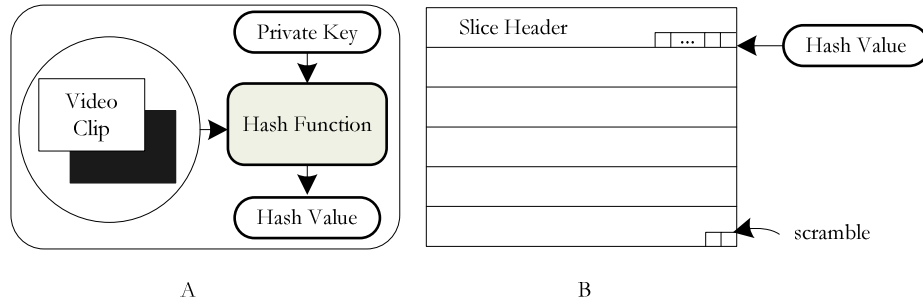


FIGURE 11. A. COMPUTING THE HASH VALUE, B. EMBEDDING AUTHENTICATION BITS IN SLICE

Insert authentication bits to the marking space. For hard authentication, watermark data is embedded as a fragile watermark. If slice header is considered to embed the authentication bits, then the frame rate after adaptation plays a significant role, which will be discussed briefly in Section c.

Optional second level authentication. After embedding the authentication bits, an optional second level of authentication can be applied by scrambling the individual frame data or slice payload to restrict re-computation of the signature by an intruder even though a private key is being used (Figure 11.B). A predefined palette can be used in this regard so that scrambled portions can be reinstated for re-computation of the hash value and/or for viewing.

b. Watermark Detector

The watermark detector consists of a 4-step process. The first step, parsing adapted gBSD, is for extracting the marking space from the adapted gBSD to identify each watermarked segment. The second step, restore frame data, comprises of restoring the scrambled frame data or slice payload bits for computing original hash value. The third step, watermark extraction, extracts the authentication bits from the slice header. The final step corresponds to comparing the computed hash value from the slice data with the extracted value from the slice header. To verify a received video content, the user needs the private key, palette (if there is one), and the adapted gBSD in addition to the video data. Secure transmission of these data is beyond the scope of this research.

c. Implementation

In the adaptation engine, while adapting the video bitstream, the authentication bits are embedded in the slice header VLC byte align bits. However, this marking space can be further extended to other entities like frames and macroblocks based on the gBSD details. While transforming the video bitstream for adaptation, a gBSD parser scans each line from the adapted gBSD. *syntacticalLabel*, *start* and *length* values are tokenized by each field according to their size from the line buffer.

TABLE 1. SAMPLE VLC_{NT} BITS AFTER ADAPTATION

Adapted Frame Rate	Worst Case (1bit/slice)	Average Case (3bits/slice)	Best Case (7bits/slice)
1	20 bits	60 bits	140 bits
3	60 bits	180 bits	420 bits
5	100 bits	300 bits	700 bits
10	200 bits	600 bits	1400 bits
15	300 bits	900 bits	2100 bits
Captured video: 30 fps, SQCIF; Capture Time: 20 Sec.; Total frames: 600			

From the gBSD, the length of the slice header VLC byte align bits (vln) of each frame is extracted and the summation of that for each video clip is calculated ($vln_T = \sum vln$) and stored. Total number of bits in a slice (S_N) is the sum of slice header ($S_{H(N)}$) bits and slice payload ($S_{P(N)}$) bits, denoted as, $S_N = S_{H(N)} + S_{P(N)}$, where N = total number of bits. For live video streams, if we take 20 second video clips of size SQCIF at 15 frames per second (fps) for processing, then if no

adaptation is required then for each clip, in the best case (i.e. 7 vlcN/slice) we shall get 2100bits to embed the signature bits for that video clip. In the worst-case scenario (i.e. 1 vlcN/slice), we shall get only 300bits. If adaptation is applied, then the lower target frame rate will reduce the size of the marking space, which is illustrated in Table 1.

The second step is to compute the hash value (F_{Hash}) based on an external private key and the video data. The architecture applies a simple hash function based on PJW Hash [89] which can be replaced by any available advanced hash function. Input to the hash function is the video data (except the bits where the authentication bits will be embedded), length of data (in bytes), and a private key (P_K). For implementation purpose, a logo/sample image of an arbitrary length (L_N) is considered to be the private key. The length of the resulting signature is restricted to the size of $vlcN_T$. Authentication bits are then embedded in the $vlcN$ bits accordingly. If the marking space is too short (e.g. less than 100 bits) and computed authentication signature is larger than that, then the rest of the signature bits can be used as a temporary private key for the next video clip. The embedding process can be convoluted using small random operations like XOR-ing the signature bits with the private key while inserting in the slice as denoted below:

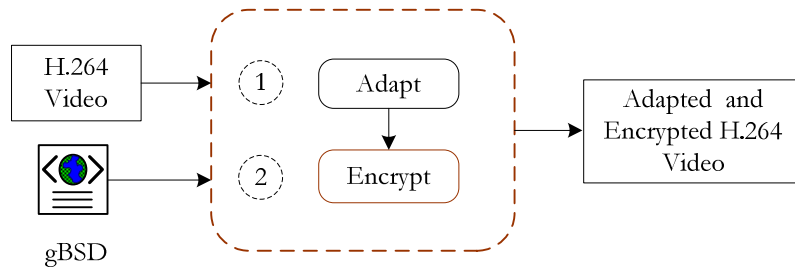
$$S_{H(N - vlcN + j)} = F_{Hash(j)} \oplus P_{K(i)} \text{ where, } 1 \leq j \leq vlcN, 1 \leq i \leq L_N$$

The final step is to apply the optional second level authentication by scrambling the slice payload to restrict the hash value calculation for that frame. In H.264, blocks and macroblocks are not byte-aligned, so XOR operation is applied to the last byte of slice payload (S_{lbsp}) with respect to the private key like that of the slice header. The modified slice payload will add another layer of assessment to detect possible attacks. Modification made to the slice payload can be shown as:

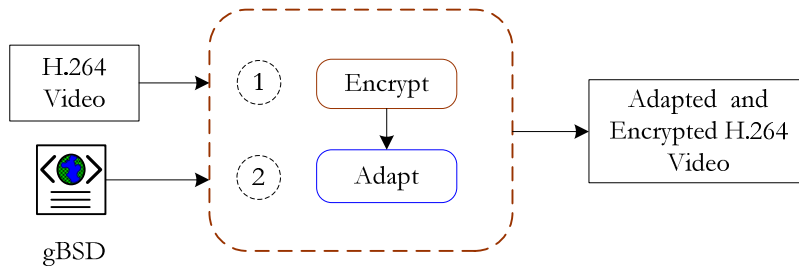
$$S_{lbsp} = S_{lbsp} \oplus P_{K(i)} \text{ where, } 1 \leq i \leq L_N$$

4.2.2. Video Encryption

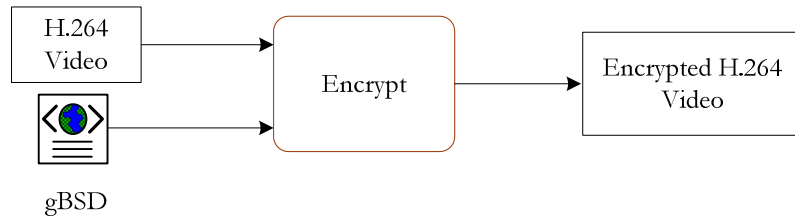
In this section, a slice-based encryption scheme for H.264 videos is detailed. The encrypted video is resilient to spatiotemporal adaptation. All the encryption steps are performed without any cascaded decoding and re-encoding operations, where gBSD is utilized to execute all the necessary footsteps in the compressed-domain. The proposed encryption scheme can be applied to a compressed bitstream before or after adaptation operations and the resultant bitstream conforms to H.264/AVC specification.



A. Step-1: Adapting, Step-2: Encryption



B. Step-1: Encryption, Step-2: Adapting



C. Encryption without Adaptation

FIGURE 12. COMPRESSED-DOMAIN ENCRYPTION USING GBSD

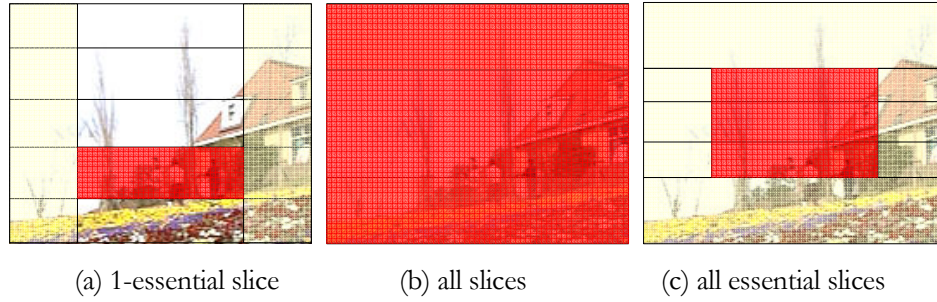


FIGURE 13. SPATIAL ADAPTATION RESILIENT ENCRYPTION

To perform the encryption in an intermediary node, it is essential to have the (adapted) compressed video along with its (adapted) metadata so that the encryption operations can be applied before or after adaptation, or even without any adaptation operations as illustrated in Figure 12. Now, to encrypt a coded bitstream, the first step is to select the encryptable plaintext with a known length for each logical unit. For this encryption scheme, slice data payloads in each frame are the logical units. Due to the slicing strategies defined in Section 4.1.2, slices can be encrypted flexibly within or outside of a region of interest (ROI). The ROI either can be an area marked by disposable slices or essential slices. The encryption module encrypts the slices depending on the user preference. If the encryption preference is the highest, then all the slices in a video frame are encrypted; else, (some or all) essential/disposable slices within the frames are encrypted to hide information.

In the encryption-module, frame and slice markers are scanned from the gBSD first. Thus, slices' starting position and corresponding length is retrieved from the (adapted) gBSD for those frames, which need to be encrypted. To encrypt the slices, an 8-bit encryption key is chosen and XOR operation is performed with the slice payload. After the XOR operation, bitstream is processed for H.264 format compliance. It is worth mentioning that each logical unit can be encrypted independently, and any encryption algorithm can be applied to replace the simple XOR operation. In any case, encrypted slices are resilient to both spatiotemporal adaptation operations. From Figure 13, we can see that if the essential slices (or all slices) are encrypted then after spatial adaptation, the adapted frames can be fully recovered by decrypting

the encrypted slices regardless of the fact that the disposable slices were discarded. To decrypt the encoded frames/slices, the end node player/receiver should have the decryption engine embedded. The operation inside the decryption is just like that of encryption module.

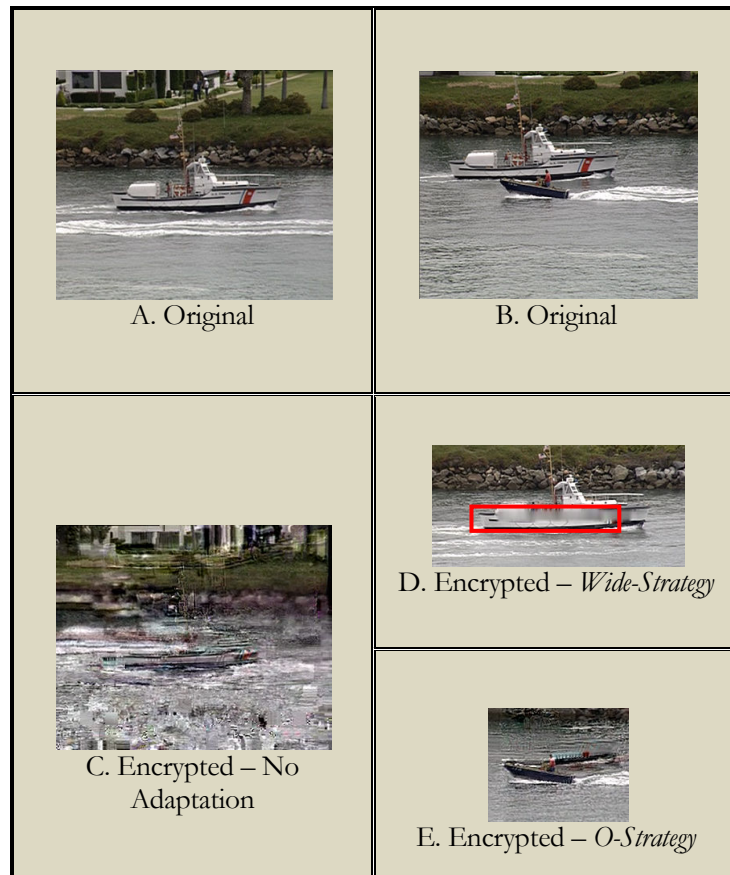


FIGURE 14. SAMPLE VIDEO OUTPUT AFTER ENCRYPTION

In Figure 14, sample outputs of the resultant encrypted frames from the test video sequence “Coastguard” are shown. Figure 14.A. and Figure 14.B. show 2 original frames from the coded video. In Figure 14.C., encrypted output of Figure 14.A. (without any adaptation) is shown where all the slices are encrypted. In Figure 14.D., an encrypted frame is shown where the encryption is applied after spatial adaptation using the Wide-Strategy. In this case, a small slice is

selected for encryption to hide information in a region of interest. Finally, Figure 14.E. depicts an encrypted version of the Figure 14.B. where encryption is done after adaptation using the O-Strategy. Since decoding is less complex than encoding, so the overhead added to decrypt the encoded frames will not exceed the nominal threshold value for presentation as they will reside as closely coupled tasks.

4.3. Performance and Results

To measure the performance and applicability of the compressed-domain video processing schemes, a compressed-domain spatiotemporal adaptation, authentication, and encryption prototypes are implemented. To evaluate the authentication and encryption performance, the compressed-domain video adaptation performance is first measured. Afterward, the time required to insert authentication and encryption data into the compressed video is evaluated. Finally, the performance of the 3-in-1 system is presented. To generate the gBSD, the ITU-T H.264 video reference software implementation [87] is enhanced with the gBSD generation functionality. The gBSD for each video is generated during the encoding process of the raw (YUV) bitstream. An Intel P4 3.4Ghz 1GB RAM computer running Win XP Pro SP2 operating system has been selected as the media processing server. To evaluate the applicability of the compressed-domain video adaptation, a cascaded adaptation module is also implemented. In the cascaded approach, the decoder available in the ITU-T sample implementation is used to decode the video; then the adapted video is re-encoded after the necessary adaptation (i.e. frame dropping to achieve target frame rate and frame cropping to achieve the spatial resolution).

4.3.1. Experiment Results for Compressed-domain Video Adaptation

Table 2 shows the Digital Item generation performance for pre-coded video adaptation. From DI generation performance, we can see that for live video streaming, QCIF and SQCIF resolutions are suitable due to their low compression and metadata generation time.

TABLE 2. DIGITAL ITEM GENERATION PERFORMANCE

Resolution	Video-1: Airshow	Video-2: Carphone
CIF	99.190s (@3.02fps)	94.893s (@3.16fps)
QCIF	23.640s (@12.69fps)	23.169s (@12.95fps)
SQCIF	11.976s (@25.68fps)	11.628s (@25.23fps)
Frame Rate : 30 fps, Intra Period: 9, Total Frames: 300, QP = 28		

TABLE 3. SPATIOTEMPORAL ADAPTATION PERFORMANCE: PROPOSED VS. CASCADED

Resolution	New FPS	Proposed	Cascaded	Proposed	Cascaded
		Video-1: Airshow		Video-2: Carphone	
CIF to QCIF (30 fps)	5	0.60s	38.48s	0.63s	39.62s
	10	0.99s	45.61s	1.14s	46.15s
	15	1.48s	64.37s	1.51s	66.87s

Table 3 presents the compressed-domain spatiotemporal adaptation performance compared to that of a cascaded approach. From Table 3, we can see that even though compressed-domain adaptation time differs for different video sequences, gBSD-based compressed-domain approach significantly outperforms the cascaded approach.

4.3.2. Experiment Results for Compressed-domain Video Authentication and Encryption

Figure 15 shows a comparative analysis of the 3-in-1 system. In Figure 15, the performance of the authentication and encryption modules on top of the adaptation module for pre-recorded videos is presented. The additional time required to embed authentication bits

depends on the hash function employed. More complicated or robust hash functions will require higher execution time to compute and embed the authentication bits. Another factor that affects the execution time is the marking space. To make the system more robust, one can decide to embed the authentication bits in the frame, macroblock, and block, which will eventually require longer watermarking time. For encryption, all the slices are encrypted regardless of any ROI in order to show the maximum time required to encrypt the test sequences. To perform encryption (and/or adaptation) operations in intermediary nodes, it is important to transmit the corresponding (adapted) gBSD together with the (adapted) video bitstream. In this case, once a specific video request is made, both the files (i.e. video and gBSD) need to be transmitted from the media resource server. To decrypt the encrypted bitstream, the decryption module of the end node needs to have the adapted gBSD to identify the encrypted slice partitions along with the decryption key.

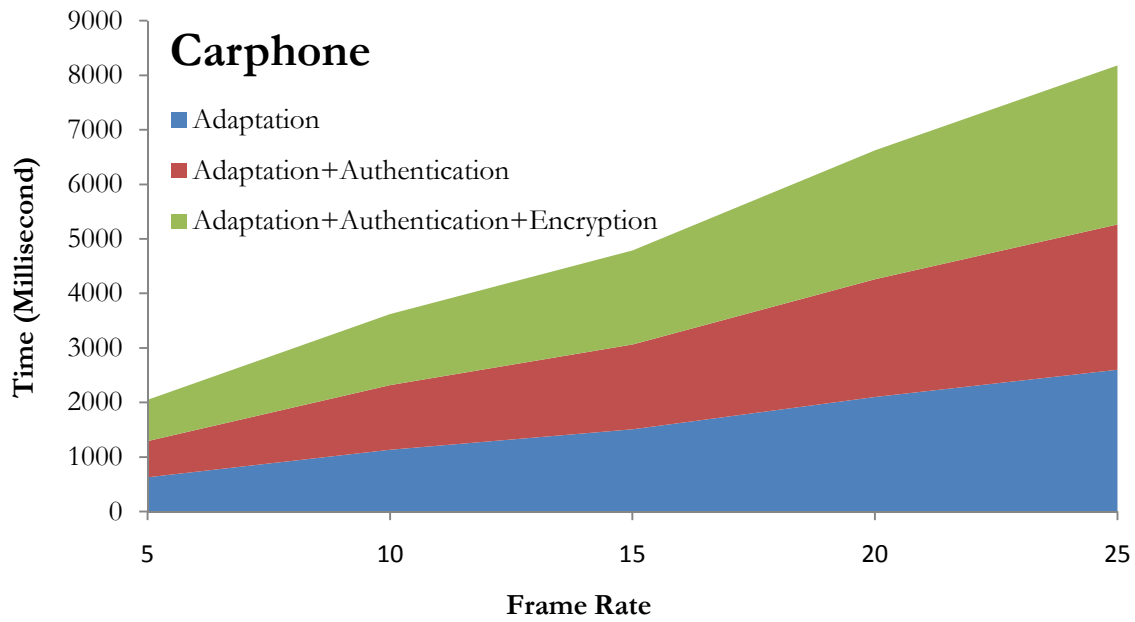
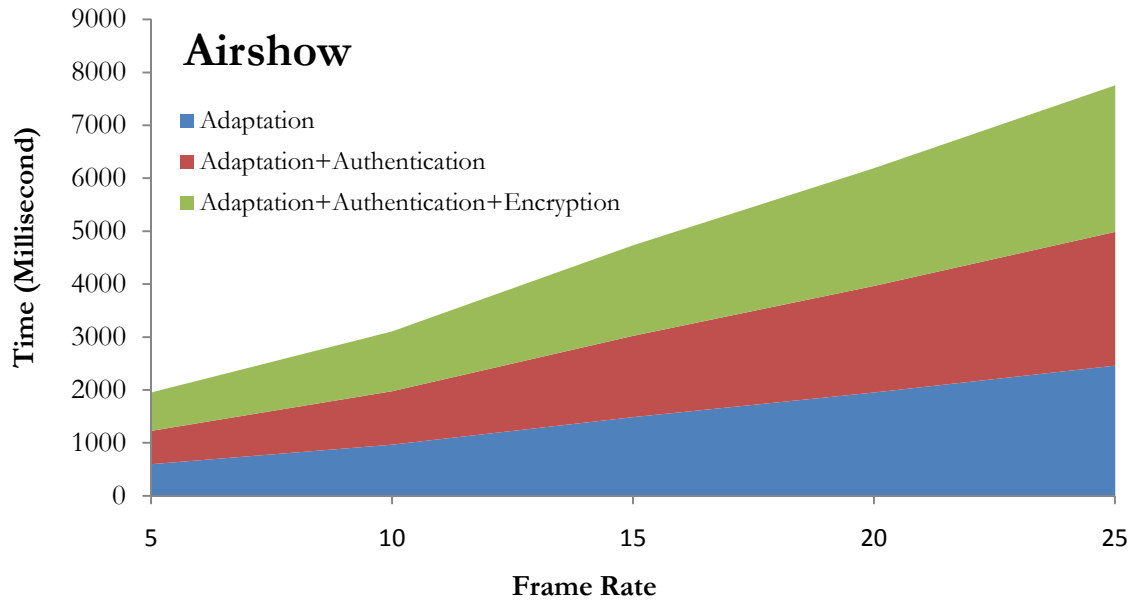


FIGURE 15. PERFORMANCE OF THE 3-IN-1 SYSTEM FOR PRE-RECORDED VIDEOS

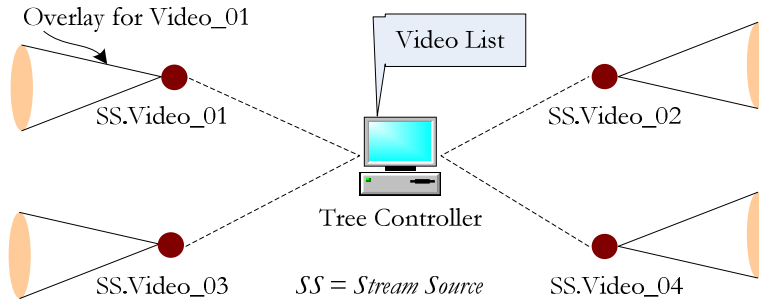
CHAPTER 5

PROPOSED FEDERATED VIDEO ADAPTATION AND STREAMING APPROACH

Given the global knowledge of the participating peers and plentiful of bandwidth and computing power in a P2P network, designing the video distribution overlays would be relatively straightforward. However, in a real scenario, designs are constrained by the limited bandwidth and computing power available with the peers. This chapter presents a simple yet scalable overlay construction, which organizes video streaming sessions among the participating peers following a master-sender-driven approach with the help of cooperative peers. The goal of this design is to distribute video contents in a P2P overlay characterized by two key factors – primarily, heterogeneous environment and devices where peers make unequal contributions to the overlay, and secondly, insufficient resources for peers to receive the full source rate. This design is a novel solution to distributed adaptation and streaming utilizing the MPEG-21 framework. The architectural design incorporates simultaneous adaptation and streaming of video contents in which a streaming session might have more than one parent nodes to serve a peer.

5.1. Adaptive Video Streaming

The proposed design is a combination of infrastructure-centric and application end-point architecture. The infrastructure-centric architecture refers to a tree-controller, which is responsible for tree/overlay administering and maintenance, while the application end-point architecture refers to video sharing, streaming, and adaptation by the end users.



A. OVERLAY GENERATION

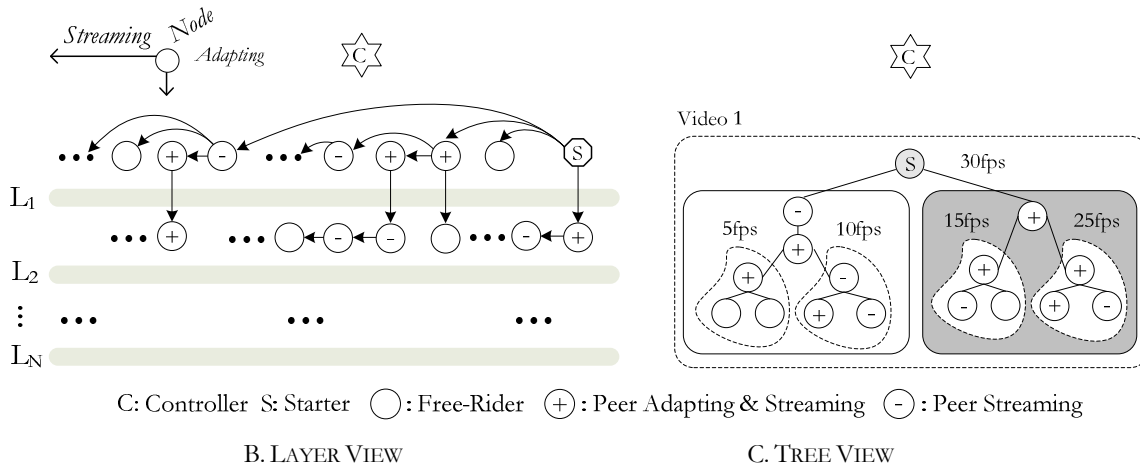


FIGURE 16. ADAPTIVE VIDEO STREAMING

Figure 16 illustrates the conceptual adaptive video streaming architecture. As can be seen from Figure 16.A., a separate overlay is formed for each video by the tree-controller. Figure 16.B. and Figure 16.C. show the layer view of the design and the tree view of the design, respectively. From Figure 16.B., we can see that a peer streams a video to other peers on its left. A downward arrow implies the adaptation operation to serve a peer requesting a low quality video. Conceptually, the number of layers is determined by the number of stream qualities. For example, if we consider adaptation requests for temporal adaptations only, then the system can have a maximum of 30 layers as long as the original streamed video is encoded at 30 frames per

second (fps). As illustrated in Figure 16.C., peers are organized into multiple sub-trees, based on a hierarchy of adaptation and streaming requirements. Therefore, to build an overlay atop this hierarchy, a set of rules are defined in the following subsections.

A tree-controller, which is the central rendezvous point, is responsible for peer joining and maintaining a multicast tree for each video stream. To deploy this architecture, a tree-controller does not need extreme bandwidth or computing power, which is usually the case in existing streaming or adaptation servers. Stream-starter, which can be an ordinary peer or any other media source, is the source for a specific video. It is responsible for encoding the video and generating the corresponding metadata. Stream-starter initiates the video streaming and forms an overlay with the help of the tree-controller. In addition to the desired video content, the stream-starter should be capable of adapting and providing overall streaming service to the immediate peers in the tree.

Each peer node is equipped with network connectivity, and able to receive and send video stream over the network. However, to make a successful P2P streaming overlay, efficiency of the overlay is directly dependent on peers' contributing bandwidth and CPU power. Therefore, the following assumptions have been made:

- A participating node will share its bandwidth as well as CPU power in the presence of some incentives, for example, peers with higher share-ratio⁵ will get priority to join an overlay.
- Each end host has limited interface bandwidth and CPU power, which eventually constrains the overall out-degree of that peer.
- Address of the tree-controller is known in advance. Tree-controller acts as the rendezvous point and assumed not to fail.

⁵ Share-ratio is calculated from the total amount of data uploaded over total amount of data downloaded

- The address of the video source (i.e. stream-starter) is known to the tree-controller only, and the tree-controller itself does not initiate or provide any video stream.
- Small hand-held devices are free from any adaptation/streaming operation and tree maintenance. They can join the overlay just to receive the adapted stream. These nodes are denoted as ‘free rider’.

5.1.1. Selecting the Number of Parents to Serve a Node

Like any P2P content distribution system, we need to decide beforehand that how many parents will be assigned to serve a node request. To serve a node by one parent, the selection process is quite straightforward where the controller needs to select a suitable parent from the existing set of capable peers. However, a single parent based overlay suffers from two drawbacks – primarily, poor resource usage and unfair contributions because a leaf node cannot contribute upload capacity, and secondly, due to the dynamic nature of nodes, the departure or failure of a top tier node can cause significant program disruption and requires frequent refinement of the overlay. Now, serving a peer with more than one peer is conceptually easy but practically, the system becomes complicated in terms of implementation and tree maintenance since there is a necessity for continuous coordination among the parents/senders. This is one of the key challenges especially when adaptation of the bitstream is a priority. Multiple parent-based systems, however, maximize the overall resource usage and system efficiency by serving peer service requests partially, if not in whole. In fact, in a P2P system, it is better to let a peer receive video clips from multiple higher layer peers because a single parent may not be able or willing to contribute the required outbound bandwidth or CPU in order to serve another peer completely. The multiple parent approach, therefore, reduces the dependency of a receiving peer on a particular parent node, which subsequently reduces the impact of the nodes’ departure. Additionally, it allows utilizing the resources with fine granularity. For example, if a peer needs 15Kbps bandwidth to serve a request but it has 5Kbps available, then it will not be able to completely fulfill the request; whereas, it could offer some bandwidth to partially respond to that request. A similar scenario may also occur for CPU power. If a peer requires 2 seconds to adapt a

frameset, while the operation needs to be performed in less than 1 second to meet the maximum delay constraint, then it is practical to schedule the CPU so that it can process and forward the clip after every 2 seconds. In the meantime, other peers will forward the rest of the clips after performing the same adaptation.

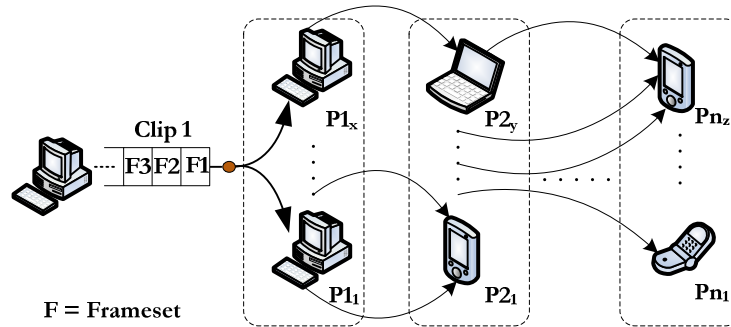


FIGURE 17. PARENT SELECTION AND FRAMESET DISSEMINATION IN THE MULTICAST TREE

TABLE 4. PARENT SELECTION

Parent Selection Name	Description
Single parent approach	A peer is serviced by at most one parent
Dual parent approach	A peer is serviced by at most two parents
Multiple parent approach	A peer is serviced by more than two parents

Now, in the adaptive streaming design presented in this thesis, the nodes at the top of the overlay are probably the wealthier nodes. However, as we can see from Figure 17, the first layer peers ($P1_1, \dots, P1_x$) will be served by a single parent (i.e. the stream-starter). Starting from

the second layer, the controller attempts to assign multiple parents for each incoming peer. If the peers are served by a maximum of two parents, then the framesets are forwarded as a coordinated odd-even frameset by the sender peers. In Table 4, we organize the parent selection approaches for future reference. The video adaptation and streaming overlay construction design to be described below allows multiple parents to serve a peer and takes the resource contribution of a peer into account to allocate the video quality to that peer in order to be fair to all the participating peers.

5.2. Video Preparation and Buffering

Considering the compressed-domain adaptation approach described in Chapter-4 as a utility tool, the way video streams can be adapted, transmitted, and buffered to match the varying network condition in an overlay is briefly explained in this section. Now, in federated video processing and streaming, the following three essential components need to be considered – i) Preparing the video, ii) Data scheduling and Buffering, and iii) Overlay construction and Streaming. In the followings subsections these components are described.

5.2.1. Processing the Video

For compressed-domain distributed adaptation in any intermediary node, all the streamed video content should have a corresponding metadata component in the form of MPEG-21 gBSD. For live video streaming, the video feed is processed in small clips as mentioned in Section 4.1.3. These video clips are encoded in the H.264 format. Simultaneously, the metadata descriptions of those clips are generated while encoding. Each clip and its corresponding gBSD are assigned a sequence number to represent its playback order on the receiver's side. For ease of understanding, in Figure 18, a coded video sequence is shown. As can

be seen from the figure below, the video is divided into clips and framesets⁶. The chain of temporal dependencies between the frames⁷ is also shown in Figure 18.

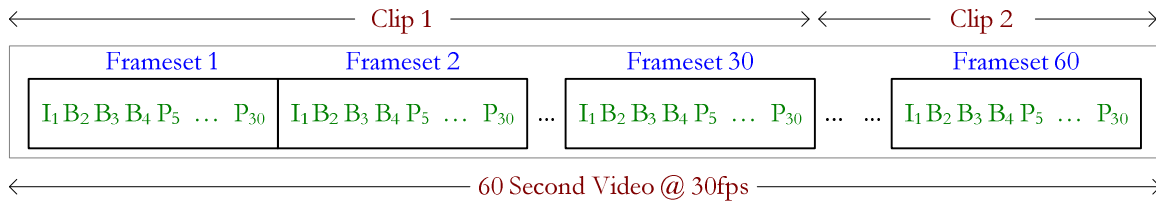


FIGURE 18. PREPARING VIDEO FOR TEMPORAL ADAPTATION AND STREAMING

5.2.2. Video Buffering

One of the major difficulties in video streaming application is to provide an uninterrupted viewing experience in the presence of network dynamics. The actual resource and environmental characteristics such as processing power and bandwidth may vary significantly between bitstream fragments during streaming. Therefore, an initial buffering (usually in the order of seconds) enables the receivers to fill up with the received video frames for continuous playback and retransmissions to another node as shown in Figure 19. The size of the buffer depends on the average frame size, number of frames in each frameset, available memory, and delay. As we proceed with the framesets, the decoder and player deals with the number of frames defined in each frameset (i.e. 1 – 30 frames). Since the live video is processed in small clips, a loosely coupled synchronization between peers is required as discussed in Section 4.1.3.

⁶ *Frameset* refers to the number of frames equal to the frame rate at which the raw video is being encoded (usually 30fps). After adaptation for 15frames per second (fps), there will be 15frames in each frameset of that clip/video.

⁷ I = Intra Frame, B = Bi-directional Frame, P = Inter Frame

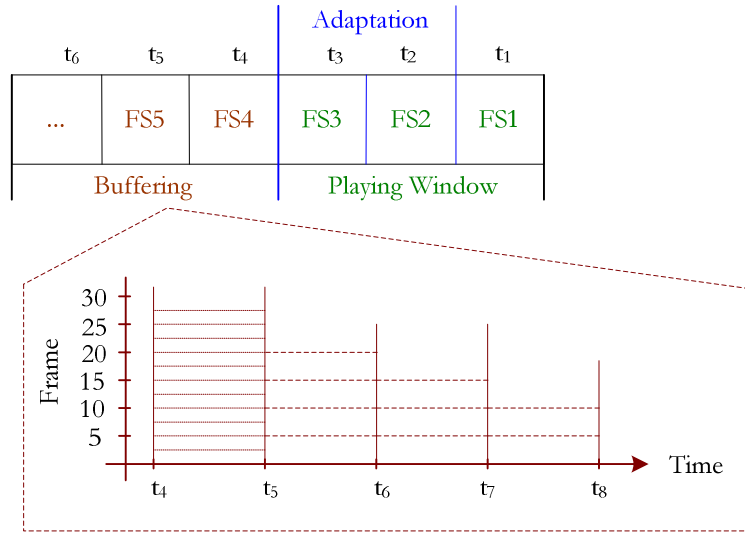


FIGURE 19. VIDEO BUFFERING, ADAPTATION, AND PLAYING IN LIVE ADAPTIVE STREAMING

On the receiver's side, stream startup time shall vary mostly due to the adaptation time variance at the intermediate nodes and the inconsistent communication delay between peers. Moreover, for live video streaming, there will be an additional delay for initial stream capturing and pre-processing. Now, if all the video segment lengths for a video stream are fixed, and the difference in processing time is small, then an initial buffer size (*IBS*) for each times units (in seconds) can be computed as:

$$IBS = \text{number_of_Framesets} \times \text{frame_Rate} \times \text{average_Frame_Size}$$

However, user's buffering capabilities will vary due to the presence of heterogeneous devices in the mobile environment and their respective memory sizes. If we consider small handheld devices which have limited memory space, the clip length can be shortened, for example, 5 second long SQCIF (128×96) video at 5 frames per second (fps). Since the gBSD is transmitted first, the *IBS* can be derived directly from the gBSD because gBSD consists of the necessary information like number of frames, frame size, etc.

On the receiver side, framesets occupy the buffer and thus, a sliding-window is introduced. As shown in Figure 20, a sliding window represents the active buffer portion of each peer. While viewing, a sender peer forwards the framesets from the active buffer portion to its immediate receiver peer(s), and adapts the frameset before transmission if required. For a peer, if the sliding window detects too many missing framesets in the active buffer portion up to a pre-determined limit, then either the administrative tasks (e.g. checking existence of the parent) begin or the video is paused and buffering continues until the number of missing framesets drop to the acceptable limit.

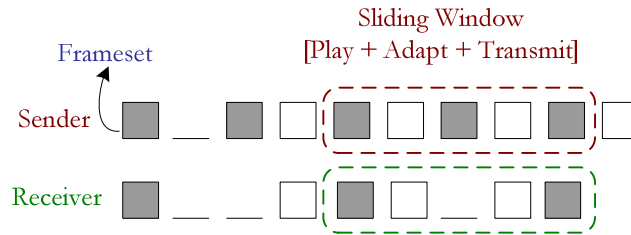


FIGURE 20. SLIDING WINDOW FOR SIMULTANEOUS ADAPTATION AND FORWARDING

Until now, it has been described that the stream-starter split live video streams into parts and then, these parts are transmitted into succession down the tree. It enables the receiver to decode and playback the content, while the video is still being delivered and buffered at the receiver. Now, if a receiver node requires an adapted video, then the sender/parent node adapts the video parts (i.e. clips/framesets) based on the gBSD before transmitting those parts. Therefore, in addition to streaming the video content, we also require the transmission of the metadata (i.e. gBSD).

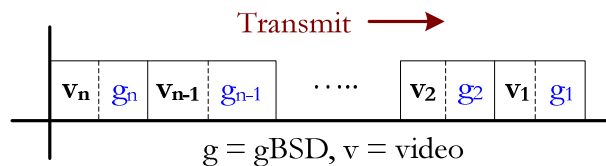


FIGURE 21. gBSD TRANSMISSION WITH VIDEO

Transmission of gBSD basically consists of the following steps (as shown in Figure 21):

1. Generate the gBSD of the compressed video,
2. Tag the gBSD with a sequence number analogous to the video segment,
3. Transmit the gBSD along with the video segment.

5.3. Overlay Construction

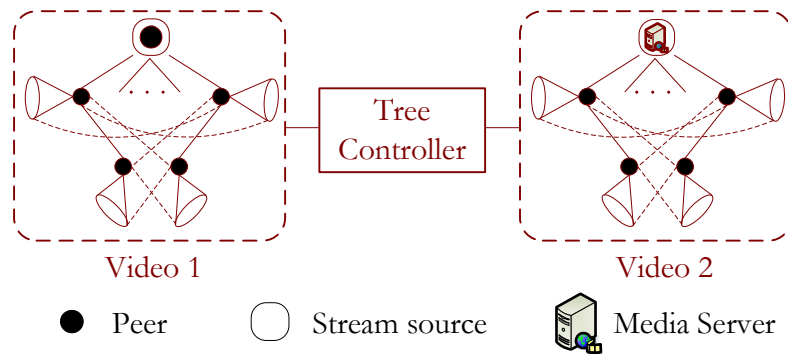


FIGURE 22. OVERLAY CONSTRUCTION: SOLID LINES AND DOTTED LINES REFERS TO PRIMARY SENDER AND SECONDARY SENDER, RESPECTIVELY

Complexity of the overlay construction depends on the number of parents to serve a peer because simultaneous adaptation and streaming need to be considered. To serve a peer, finding more than one parent having enough CPU power and bandwidth is sometimes inconvenient, especially when the heterogeneity of peers is taken into consideration. In this proposal, a maximum of five simultaneous parents are assigned to serve a peer provided the fact that not all the parents may have similar resources. Figure 22 illustrates the concept of the overlay construction. In this design, either a media server or a participating peer may initiate the streaming and form an overlay with the help of the tree-controller. This means that for each video, a separate unidirectional single source overlay is formed and all these overlays are managed by the tree-controller. Each video stream originates from the stream-starter, which is the root

node for that particular overlay. It is possible that a stream-starter may not have enough CPU power to do adaptation but the starter should be able to provide streaming service to some immediate receiver peers in the tree. Moreover, to perform adaptations in intermediary nodes, a stream-starter must be able to produce and forward the corresponding metadata component in the form of MPEG-21 gBSD for the streamed video content. The tree-controller is not responsible for providing any video stream itself. Small hand-held devices are out of scope from any adaptation operation and tree maintenance. They can join the overlay just to receive the (adapted) stream. In addition, it is optional to transmit gBSD to these devices since they are not capable to adapt video. From Section 2.2 and Section 4.1, we have learned that the tree-controller gathers information about an incoming peer's computing power or bandwidth from UED which include the description of terminal capabilities (e.g. codec capabilities), network descriptions (e.g. maximum or minimum bandwidth), user characteristics (e.g. personalized requirements), etc.

5.3.1. The Tree Controller

The tree-controller acts as a centralized index of the available streams and stream sources. It authorizes, classifies, and clusters peers according to their bandwidth and CPU power when they attempt to join an overlay. The tree-controller monitors the status of peers as well as maintains the dynamics and the diversity of the overlays as reflected in the following cases: i) A sender may stop contributing by hanging, and ii) Out bound bandwidth and computing power may change over time. The tree-controller keeps track of all the connected peers to the overlay(s). A set of nodes, Set_p , served by another peer, P , is determined by the controller based on the selection procedure presented in Section 5.3.3. P receives the updated list and required information from the controller. If the stream-starter leaves or fails, then the tree is dissolved immediately by the controller because stream-starter was the source of the video content for that tree.

From the UED, the tree-controller computes the adaptation requirements. For example, if there is no preference from the recipient, then depending on the downloading bandwidth, the temporal adaptation requirement can be computed. Otherwise, the controller attempts to serve

the node according to its preference. However, a node's video requirement cannot be higher than the original streamed video. The tree-controller itself stores the peer information as well as sends a copy of the graph to the stream-source. Therefore, stream-source is the secondary administrative point of contact if the tree-controller fails.

In this design, backup parents are not considered. Peer information of each overlay remains sorted by the tree-controller to ease the assignment of a parent in the case of a sudden node departure. Consequently, the tree-controller (re)-assigns a new peer as a parent when a parent leaves the system. However, to serve a disconnected peer, the tree-controller maintains a node priority. For example, existing peers have higher precedence over new incoming peer and free riders.

5.3.2. Peer Registration

A new peer that has not yet joined a video overlay contacts the tree-controller and gets a list of the available streams. The tree-controller registers the peer for a particular overlay and sends the parent information. Based on the incoming peer's requirements, the controller selects parents that can serve this peer possibly through the shortest (i.e. smallest height) path in the tree. An incoming peer may be served by one single or more parents depending on the resource availability.

Every incoming peer passes basic information entailing device requirements, network characteristics, adaptation capabilities, and personal preferences in the form of UED to the controller. From an implementation view point it can be seen like this – every peer wishing to join this system will have a program running in its end. The program will register the peer to a multicast streaming session. The idea is that the program will know the address of the tree-controller beforehand like that of Yahoo or MSN instant messenger. However, if the system is deployed within a neighborhood then the address of the tree-controller can be set manually. To enable an automated peer discovery, the Peer Discovery Protocol (PDP) within JXTA services [97] can be applied. Once connected, the peer will receive a list of available streams. Then the

user will choose a stream to view, based on which, the tree-controller will assign parent(s) to that peer.

To compute CPU power of a peer, the tree-controller needs specific information such as processor speed, L2 cache, memory, clock rate, etc. Nevertheless, absence of a benchmark to classify these computation factors make it complicated and unreliable if we take into consideration the diverse hardware/systems available these days. To overcome this issue and for implementation convenience, the CPU power is mapped into adaptation time and so, an adaptation timing profile is created for each peer. Towards this, a peer will adapt a small video clip for different spatiotemporal adaptations while installing the program. Based on this adaptation time, the controller will be able to know how many adaptations this peer can perform in a certain period, and assigns this peer an '*adaptation and streaming peer*' role or a '*streaming peer*' role. Peers capable of both adaptation and streaming, or only streaming, are denoted as regular peers. Devices with limited resources are free of any streaming or adaptation operations and may join as a free rider. Mobile peers are those peers who are not physically static. They can be a regular peer or a free rider.

In Figure 18, we showed that framesets are used for live adaptation where each frameset refers to a one-second video consisting of 30 frames. So, for real-time adaptation, considering the initial startup delay, a capable peer has to adapt each frameset in less than one second – hence, the total number of adaptation tasks it can perform in one second will symbolize how many peers it can serve in terms of adaptation (provided it also has enough bandwidth). For example, if a peer requires 100 millisecond to adapt a frameset from 30fps to 20fps, then in one second it can perform around 10 similar adaptation operations. The tree-controller maintains an adaptation profile of all the peers. Based on this adaptation time, the controller computes the number of adaptations this peer can perform in Time-Division-Multiple-Access fashion.

5.3.3. Parent Selection

Based on the incoming peer's requirements, the tree-controller selects the parents that can directly serve this peer possibly with the smallest height in the (sub) tree. If the tree-controller is unable to find more than one parent, then it attempts to assign one parent. The tree-controller executes the following steps to select the candidate parent(s):

1. A set of potential parents, Set_1 , is selected as a stream-source that has the requested video (vid) along with the gBSD, i.e., $Set_1 = \{p \mid p \in Overlay, available_video(p) = vid_r\}$.
2. Find a subset of Set_1 based on available bandwidth, i.e., $Set_2 = \{p \mid p \in Set_1 \text{ and } available_bandwidth(p) = true\}$.
3. Check adaptation requirement: For each $p, p \in Set_2$, if there are peer(s) having the same video quality, then no adaptation is required.
4. If adaptation is required, select a subset of Set_2 based on available computing power i.e., $Set_3 = \{p \mid p \in Set_2, available_CPU(p) = true\}$.
5. Finally, the cumulative distance of each peer, $p \in Set_3$, to the controller (in terms of delay) is measured to select the parent(s) that has minimum distance to the controller.

5.3.4. Parent Synchronization

When a peer is served by more than one parent, then the controller assigns a number for each parent. Eventually, a parent forwards those framesets from a video stream that corresponds to its own number to serve a particular peer. In this regard, every peer synchronizes its virtual clock with that of the stream-source. Each frameset contains originating timing information. Based on this information, each sender peer identifies which framesets to forward. On the receiver's side, received framesets are coordinated in the buffer where buffer size varies depending on the number of framesets that a peer can accommodate in its memory.

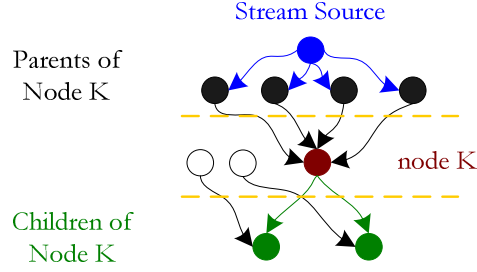


FIGURE 23. ILLUSTRATION OF ADAPTATION AND FORWARDING OPERATIONS

Using Figure 23, an example of parent synchronization for simultaneous adaptation and streaming operation is given for node K , where node K receives framesets from upstream peers, and then, forwards framesets to downstream peers. For a peer K being served by multiple parents ($PSet_K$), each of its parents selects and forwards framesets (FS_{send}) as follows:

$$FS_{send} = (PS_K + n \times TP_K)$$

where PS_K = serial number of the parent for receiver 'K', TP_K = total number of parents for 'K', $n = 0, 1, 2, \dots$, while true

Peers in $PSet_K$ declare the frameset availability, and synchronize the starting frameset as follows:

$$(\text{MaxOf}(\text{Current_FS_Playback}(PSet_K)) + 1)$$

Finally, Nodes in $PSet_K$ set an offset for framesets, and start from 0 to serve K . On the receiver's side, stream startup delay occurs due to video processing operations (i.e. video capture, encode, adaptation etc.), parent selection and synchronization, buffering, and transmission. To reduce this delay for a new peer, the controller initially assigns a parent(s) in a sub-tree, which has a stream close to the actual requirements. Gradually, video quality improves from the next framesets.

Remarks:

- The required bandwidth varies for different videos. When a peer wants to share a video, it also sends the stream related information to the controller.
- Availability of bandwidth has higher priority than available computing power because if there is not enough bandwidth, then adapted video streams cannot be streamed to another peer.
- It may seem that when a new peer joins, the algorithm has to be run again. Hence, Set_t is temporarily saved by the tree-controller to avoid re-computation every time a new peer joins the overlay. The controller updates this list over time when a peer leaves the overlay. However, the list is recomputed after tree refinement operations (Section 5.3.7).
- For the multiple parent case, each parent is responsible to deliver separate chunks of the video. Obviously, not all parents may have similar computational and network properties in the multiple parent approach. Therefore, adaptation and streaming requests served will be based on the minimum the parents can support. In the worst case, the controller attempts to assign at least one parent.
- Due to peer mobility, if the upload bandwidth of a parent node decreases and affects the bandwidth to serve its immediate peers, then it will attempt to serve peers according to their peer class (see Table 6) and joining order. It may also notify its parent(s) to reduce the bit-rate/frame-rate of the transmitted video. In such a case, the new rate is applied to the next available frameset.

5.3.5. Peer Joining Constraints

Even though we want to deal with peer heterogeneity and offer services to small handheld device, it is also true that a peer having no sending throughput and computing power does not let the overlay scale beyond that point. In such a case, the overlay will be saturated

quickly when too many receiver-only peers join the overlay at the beginning. This fact cannot be ignored and is taken into account by placing the following constraint:

- If the incoming node is a ‘receiver-only’ peer then the selected parents must be able to serve at least one more regular peer.

5.3.6. Peer Departure

In a live streaming session, it is very likely that a node may depart gracefully by informing the controller or ungracefully by hanging, leading to service interruption. Therefore, if a node is serviced by multiple parents, then the node will experience freezing of video for those framesets that were supposed to be forwarded by the departed parent. However, when the number of parents is higher (e.g. 4 or 5) then the missing frameset problem will cause comparatively less annoyance to the viewer since the frameset drop will occur after a short buffer time and it will continue until a new parent has been assigned.

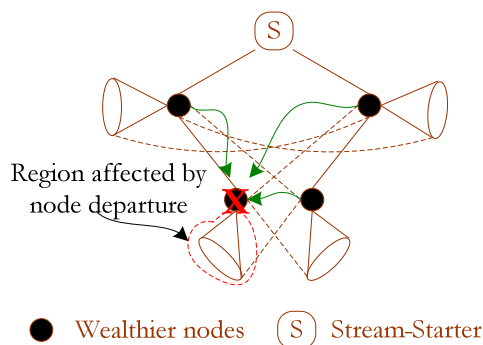


FIGURE 24. RECOVERY FROM SUDDEN NODE DEPARTURE

In this design, higher priority is given to the disconnected nodes than new node join requests. If there are multiple reconnection requests from the peers of a (sub) tree, then the controller verifies the existence of the parent of that (sub) tree. Otherwise, if there is no reconnection request, a parent reports the departure of such a node to the controller. If the source node (i.e. stream-starter) has departed the session, then that overlay is dissolved

immediately. If a root node of a sub-tree has departed, then the tree-controller attempts to assign a parent from the sorted availability list (from the same level or the ancestors of the disconnected parent). Recovery from sudden node departure is illustrated in Figure 24.

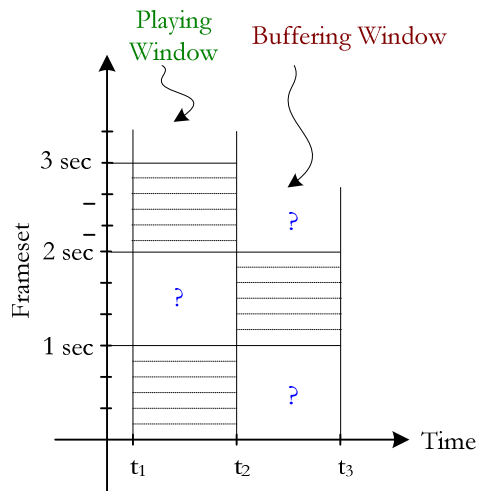


FIGURE 25. FRAMESET SKIP DUE TO SUDDEN DEPARTURE OF A PARENT

Now, in the case of a parent disconnection, if that parent's children remain connected to the overlay, then they will keep receiving the streaming service from other parents (provided that they are being served by multiple parents). The reason behind this design decision is to keep the peer disconnection rate low and to give service priority to the existing node(s). Giving service priority to existing node means that as soon as a capable peer is discovered to replace the departed parent, the controller will assign the new peer as a replacement. Until then, the controller instructs the existing parents to serve the affected node with extra resource, provided the fact that these parents have extra bandwidth and computing power. If the existing parents cannot extend its service towards that affected peers then until a replacement parent can be assigned, the affected peer will experience temporal jitter (due to frameset skip as shown in Figure 25) in the video. Now the fact cannot be avoided that sometimes there can be few good and available parent choices in the system. Therefore, if the previous parent exists in the system,

then the controller tries to re-establish the connection with the previous parent. The tree-controller applies the following steps to find a replacement parent for the affected node(s):

1. If the departing node is a leaf node without any responsibility, the controller updates the Set_i (from Section 5.3.3) and informs the parent.
2. If a peer gets disconnected and it requests a new link, then the controller checks the existence of its parent.
 - If the parent is active, then the controller instructs them to reconnect.
 - If the parent has departed and it was the stream-source, then the tree is dissolved immediately.
 - If the parent is the root of a (sub) tree, then the controller chooses a capable peer from the same level or the ancestors of the disconnected peer and instructs to serve the affected peers.
 - If the above steps fail, then the controller starts distributing the children of disconnected parent in the overlay.

5.3.7. Tree Refinement

It is understandable that due to the dynamics of both node joining and node departure, the overlay structure can easily become sub-optimal. Now, if assignments of peers do not yield the full quality vis-à-vis delay requirement of a particular session, then a peer switching (i.e. tree refinement) is attempted. Additionally, overlay reconfiguration facilitates expanding the service to new nodes. A simple refinement technique is applied by shifting the peers with higher aptitude and stability (decided from peer history) into the upper layer of the tree. The tree refinement procedure is called when end-to-end delay of certain percentage of receiving peers exceeds the predetermined threshold for an overlay. Parent switching will take place in a way that no peer starves due to this operation. The tree-controller first checks the aptitude of the available nodes,

(i.e. available bandwidth) CPU power, delay from the controller, and share-ratio of a peer. Afterwards it refines the overlay by placing the nodes that are more capable in the upper layer of the (sub) trees. For example, consider the case in which p and q are two existing peers and c is the stream-starter, where p is connected to c , and q is connected to p . Suppose that the delay of a new incoming peer s will exceed its threshold if it is served by the first available serving peer p . Therefore, the tree-controller will replace p with s , satisfying the following conditions (otherwise, p will serve s):

1. $out-degree(c) > 1$
2. $bandwidth(s) \geq bandwidth(p)$
3. $CPU-power(s) \geq CPU-power(p)$
4. $delay(c,s) < delay(c,p)$
5. $share-ratio(s) > share-ratio(p)$

Condition ‘1’ above ensures that c can serve s without starving p . Once the node c and s is connected, the tree-controller instructs p to connect to s .

CHAPTER 6

ADAPTIVE VIDEO STREAMING APPLICATION DESIGN AND PERFORMANCE ANALYSIS

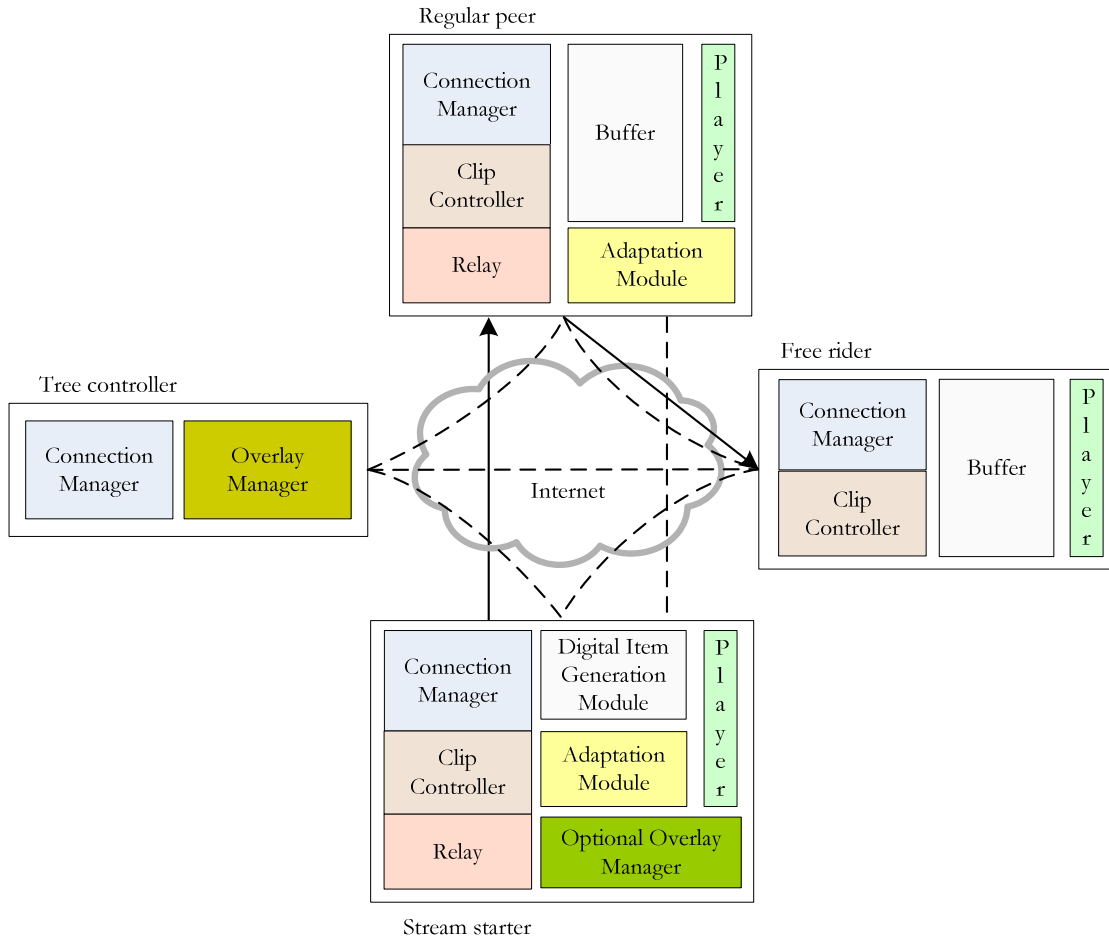


FIGURE 26. MULTI-PARENT ADAPTIVE VIDEO STREAMING DESIGN - SOLID LINES AND DOTTED LINES, REFERS TO VIDEO DATA AND CONTROL DATA, RESPECTIVELY

In online video streaming, a participating peer plays the dual server-consumer role unlike the ordinary commercial systems where clients are considered as unidirectional resource

consumers. From application viewpoint, a federated video streaming system can be seen as a pure client-server application. Therefore, the components of the proposed system design are mainly divided into receiver side functions, sender side functions, and video processing module. However, for small handhelds, the application can only have the receiver side functions. Now, while implementing, there are three software components: for the tree-controller, for regular peers, and for mobile peers/free riders.

TABLE 5. DIFFERENT SYSTEM COMPONENTS OF THE ADAPTIVE VIDEO STREAMING SYSTEM

Component Name	Presence	Description
Connection Manager	Tree-controller, Regular Peer, Source Peer, Free rider	Responsible for handling individual TCP/IP and UDP/IP connections among nodes
Overlay Manager	Tree-controller, Stream starters	Keeps track of overlay related data, e.g. peers, stream-source. Also assigns parents, handles node joining, peer failure etc.
Clip Controller	All peers	It coordinates clips and corresponding gBSD. For senders, it also tags the clips and gBSDs. It also maintains the share-ratio.
Relay	Stream-starter, Regular Peer	It forwards the received video segments (after adaptation if required) to other peers connected to this peer.
Digital Item Generation Module	Stream-starter	Generates H.264 encoded video and corresponding gBSD
Adaptation Module	Stream-starter, Regular Peer	Responsible for adapting video segments according to receiver's requirement

The tree-controller functionalities are quite straightforward - it creates and manages separate overlays for each video stream. Regular peer is capable of initiating, adapting, and streaming a video stream. Additionally, it is possible to append the tree-controller functionalities in the regular peer. Free rider is capable of receiving streamed data only. Figure 26 gives a snapshot of the multi-parent adaptive video streaming design from a client-server application perspective. In Table 5, different components, their descriptions, and their presence in the adaptive video streaming application are highlighted.

6.1. Dynamic Quality Adjustment

In live online streaming, start-up delay refers to the time gap between the selection of a channel and the actual playback starting on the screen. This is a critical performance issue for the users who do a lot of channel surfing as well as users requesting reconnections due to peer churn and/or network dynamics. In order to tackle this issue at the joining time, the joining peers are temporarily served with a video that is less than (or equal to, if possible) the requested quality and is readily available. This limits the start-up delay to the transmission of the video data and processing time at the user side.

Now let Q_i denote peer i 's attainable quality based on current bandwidth and processing ability and let QR_i denote the received quality of peer i at anytime. Upon joining a streaming session, peer i receives the information of its attainable quality, Q_i , from the tree-controller. However, to reduce the start-up delay, tree-controller assigns parent(s) to fill-up the buffer of i with the video of certain quality, QR_i , which is as close as possible to the attainable quality (i.e. $QR_i \leq Q_i$). The tree-controller starts a timer (t_i) for peer i , $t_i = rand(1, T)$, where T is a retry time period. The retry timer is set to be $t_i = rand(1, 2^{cls} T)$, where cls is the peer class (see Table 6). Upon the expiration of the (retry) timer, the tree-controller tries to assign new sources/parents to upgrade the video stream quality towards Q_i . If sufficient parents are found to serve the future video chunks with a higher quality than QR_i , QR_i is updated. Meanwhile, peer i itself runs a quality decrement process to ensure that it can cope with the expected resource contribution limit. The

quality decrement process also periodically monitors the status of available resources within the peer itself and compares the received/attainable quality with the sustainable quality at that time. If it requires updating QR_i or Q_i in order to match the resource contribution limit, then the peer informs the tree-controller accordingly. Detail of the resource contribution limit is discussed in Section 6.3.

6.2. Service Fairness

The adaptive streaming system attempts to serve an incoming peer with the video quality requested or an adapted version of the content in terms of either frame rate or spatial resolution, or both. The provided service, however, depends on the available resources. In resource-scarce environments, we must consider utilizing any available resource that can partially fulfill the demand. Even though the performance should be consistent over time, it is acceptable to respond to a peer's connection request by partially fulfilling the peer requirements in order to reduce denial of service. Now, the service quality should be degraded uniformly if there is a lack of network and CPU resources. Since peers have different quality and streaming requirements depending on their viewing device, providing the same quality to all the peers is unfair in terms of the perceived quality. Therefore, the notion of fairness is more subtle. We emphasize on service fairness, which refers to serving a peer request by allowing server node(s) to offer at least a certain percentage of the requested service when there is resource scarcity. In order to satisfy this goal, a global service fairness constraint, S , is introduced for all the nodes in the system for a particular overlay. For example, if a peer needs 50Kbps bandwidth from the system, but receives 25Kbps, then $S=0.5$. Therefore, S can be considered a global fairness constraint for the entire overlay to determine the ratio between the quality of the content delivered and the quality of the content initially requested. S ensures a similar service quality to the peers, independent of their viewing environment as well as to minimize the possibility of denial of service to an incoming join request. The main advantage of imposing this constraint is to maximize the overall response by enabling equal service distribution to all the participating peers and ensuring fairness in that way.

6.3. Peer Contribution

There is no doubt that the performance of a cooperative video streaming system is highly dependent on the peer resource contribution. In [90], authors introduce a “contribution-aware” policy where the basic concept is that a peer is entitled to receive more if it contributes more. Motivated by [90], a resource contribution policy is implemented which is slightly different than that of [90]. In this policy, a peer is first allowed to receive its requested rate, and then the peer is required to contribute resources in the system based on its receiving rate. Obviously, contributions from the peers are most likely to be heterogeneous due to the heterogeneous environments in the real world. Moreover, many high-speed internet users have asymmetric connections (i.e. large downloading capacity than uploading). Therefore, a resourceful peer may need to contribute more compared to a resource-limited peer in order to keep the system running and to ensure that there is extra bandwidth and CPU power in the system. Inevitably, an optimal overlay requires the resourceful peers to contribute more whereas those peers may choose to participate selfishly. To address this issue, an effective taxation-based “minimum contribution” requirement is formulated. Minimum contribution requirement ensures that resources allocated to serve a peer are proportionate with that peer’s contribution rate. The goal is to enable peers to ensure different levels of contributions of resources based on their received quality while effectively utilizing bandwidth and CPU resources available in the system. The formulation draws a direct mapping among the actual amounts of resource consumed by a peer, peer classification, and the amount of resources that the peer should contribute. Essentially, an equitable resource distribution among peers with similar contributions is facilitated.

Resource availability is computed from the total simultaneous adaptation and streaming requests that a peer can handle. For ease of computing, a weighted sum of the bandwidth and the computing power is considered to compute the resource availability of a peer. Resource availability (RA) of peer i is computed as:

$$RA_i = \text{streaming_slot}_i * 0.8 + \text{adaptation_slot}_i * 0.2$$

Bandwidth availability receives more importance than the CPU availability because of the fact that without having enough bandwidth, we cannot forward an adapted content to another peer. Now, let RG_i be the minimum resource that needs to be contributed to the overlay by the user i . Under a contribution rate, C , where $0 \leq C \leq 0.5$, the minimum RG be given by user i is –

$$RG_i = (L+C) RR_i + C * \sum (RR_i/N)$$

where $L=0.5$ for Tier-1, $L=0.35$ for Tier-2, and $L=0.2$ for Tier-3 peers

The minimum resource contribution rate, RG , for a peer i consists of two parts. The first term represents the minimum resource a peer is required to share by accepting resource RR . The second term is a matching rate aggregated from the minimum resource contributed by other peers of the same tier of peer i . L and C are global parameters that are known to all peers. The tree-controller supplies information regarding contribution rate of the other peers of the same tier.

L is used keeping in mind the practical setting of peers from each tier (see Table 6 in Section 6.6.2). L corresponds to the following two facts – i) Same tier peers should contribute equally, and ii) Less capable peers will contribute less compared to higher tier peers. Peer heterogeneity and fairness is thereby ensured by forcing each peer to contribute to an extent that matches the contribution of similar peers.

For the contribution rate, as C approaches zero, the minimum resource contribution rate approaches to half, one third and one fifth of the resource received by Tier-1, Tier-2, and Tier-3 peer respectively. Again, as C approaches one, the system impractically urges the peers to contribute a rate that is nearly double of the received rate. Therefore, we limit the value of C to be between 0 and 0.5. Usually, C 's value will be higher when there are fewer nodes in the overlay, and C decreases as the number of peers increases because the respective contribution also increases.

6.4. Handling Network Fluctuations and Packet Loss

Typically, adaptation requirements and/or bandwidth requirements are computed and set at the beginning of the streaming sessions. However, on the receiver's side, a quality decrement process also periodically monitors the resource fluctuations and re-computes its requirements. Let us consider a coded video of QCIF (176×144) resolution at less than 150kbps having each P-frame in a single packet. The attempt to transmit this video over a lossy channel will obviously result in an adverse effect on the perceptual quality of the streamed video, as discussed in [91]. Therefore, in order to send the video data over the network, video data is encapsulated in packets and then delivered over the IP network using UDP. Packet loss is identified through the sequence number at the receiver side. In case of high adaptation ratio, for example, 3 frames per second (fps), if a key frame cannot be reconstructed due to packet loss during transmission, then instead of retransmitting that frame, the application will skip that frameset and go to the next frameset on the receiver side.

6.5. Detecting Peer Failure

Peer connection failures are expected to happen often due to the unpredictable behavior of peers as well as the excessive load in the underlying network. The first step requires a peer to detect a failure between it and the parent. Then the tree-controller initiates the second phase to recover from the failure by finding a new parent. A peer failure is detected by monitoring the TCP control channel established between the receiver and each of the sending peers. If a connection reset is detected, then either the receiver peer(s) or the parent of the departed node informs the tree-controller about the peer failure.

6.6. Evaluation and Results

For the cooperative video streaming design, experimental results are collected over a simulated network. The heuristic approach for adaptive video streaming is evaluated with a view to answer the following questions:

- First, what is the real-time adaptation performance?
- Second, what is the node joining and reconnection success rate?
- Third, what is the achievable resource usage?
- Fourth, what is the impact of the minimum contribution requirement?
- Fifth, what is the impact of the dynamic quality adjustment?
- Sixth, what would be the Quality of Experience of a viewer due to peer churn?
- Seventh, what is the impact of the service fairness on resource utilization and node joining success rate?

Finally, the results obtained from the heuristic approach are compared with that of the optimum overlay in terms of achievable service fairness and its impact on resource utilization as well as node joining success rate.

6.6.1. Performance Metrics and Experiment Setup

To evaluate the proposed adaptive video streaming system design, the following performance metrics are defined:

- Adaptation Timing - If we consider delay in an overlay, then the network delay between peers is negligible with respect to the adaptation time. Therefore, in this section, an adaptation time profiling for different peer-class will be given to show the feasibility of online compressed-domain adaptation.
- Node Joining Success Rate - The major focus of this design is to increase the service quantity. Therefore, node joining success rate represents the effectiveness of the overall design. Here, node joining includes both new incoming peer request and disconnected peer rejoin request.

- Peer Stretch - Since we are interested in live video streaming, therefore, the number of hops in the ALM tree is important to measure the Quality of Experience (QoE). Higher number of hops will introduce larger delay from the source node to an end node. In order to assess the overhead of overlay streaming, the metrics peer stretch has been used, where,

$$\text{Peer stretch} = (\text{Delay along the ALM path}) / (\text{Unicast delay})$$

- Bandwidth and Adaptation Slot Utilization - Bandwidth and Adaptation slot utilization gives an idea of the portion of the total overlay resources utilized. For each peer, adaptation slot refers to the average number of adaptation operations a peer can perform in one second based on the adaptation timing profile of the peer.
- Bandwidth Service Index (*BSI*) - *BSI* is the ratio of the number of receivers that the total overlay bandwidth can potentially sustain (at a maximum rate determined by the tree-controller) versus the number of actual receivers.

BSI < 1: Not all accepted peers in an overlay can receive the maximum rate

BSI = 1: The overlay is already saturated

BSI > 1: The overlay is less constrained

- CPU Service Index (*CSI*) – *CSI* is the ratio of the total number of adaptation slots in the overlay versus the number of adaptation slots used by the receivers.

CSI = 1: No adaptation slot available

CSI > 1: Peers requiring adaptation can join

- Service Disruption Index (*SDI*) – The Quality of Experience (QoE), in terms of viewing a video stream, decreases due to the frequent node joining and node departure –

especially when these are service nodes. To evaluate the QoE by a viewer, a matrix named Service Disruption Index (*SDI*) has been introduced which is the ratio of the total streaming disruption time versus the total viewing time since the playback begins.

For our experiments, we developed a video chunk level event-driven simulator in C++ to evaluate the overlay performance. The ANSI-C clock method is used to measure the necessary timings. The trace is collected for three different video streams (two pre-coded test sequences and one live stream) involving a maximum of three stream sources and a tree-controller. The number of concurrent peers varies from 25 to 150 for each video. The maximum number of simultaneous peers for a video is restricted to 150 for the ease of comparing the results from the heuristic approach with that of the analytical approach. The stream-starter is capable of both streaming and adapting to its immediate peers. For all 151 nodes, the following properties are set – a node ID, peer-class, upload bandwidth, download bandwidth, and alive-time. For the stream-starter, the corresponding video properties such as video resolution (CIF, QCIF), frame rate (5-30fps), and average frame size (CIF: 10Kb, QCIF: 6Kb) are also set. The participating peers are heterogeneous (in terms of available bandwidth) and demonstrate random behavior (in terms of arrival, departure, and streaming/adaptation requirements). The nodes are initially kept in a pool of not-connected-nodes. Once the simulation starts, the simulator uses a random number generator to pick a node from the pool in every 5-time units. For each node upon successful connection, alive-time varies from 100-time units to 5000-time units. The end-to-end delay threshold is set to 300ms. In some experiments, batching is introduced. Batching incurs service latency but increases the possibility of larger group formation. This is because when the batch duration is introduced, similar requests are accumulated within this short interval of time.

To make the simulation realistic, the asymmetric nature of nodes has been considered. This means that nodes behind the DSL and Cable can receive several hundreds of kilobits per second but can fundamentally only donate less. The upload and download bandwidth for a peer varies from 1Kbps to 1Mbps and 8Kbps to 7Mbps respectively, which is based on the experimental conditions used in [98]. Moreover, the upload and download bandwidth changes

during the alive time of a peer in the overlay. By default, temporal video quality is computed based on the download bandwidth of an incoming node unless a specific quality is requested by the incoming node. It is also assumed that on average it requires 6Kbps of bandwidth per QCIF video frame (about 180 Kbps for a video at 30 fps) and 10Kbps of bandwidth per CIF video frame (about 300 Kbps for a video at 30 fps) regardless of the video motion and frame type. While joining, two connections are established with each incoming peer – i) A UDP connection for sending the video stream, and ii) A TCP connection for sending the control data. In an area of 10^8 sq. kilometers, all peers are placed randomly. Peer-to-Peer delay is measured computing the Euclidean distance between peers and assigning a delay per unit distance. For ease of implementation, throughout the simulation it was assumed that delay between any two peers is symmetric. The results presented here are the average results of at least 10 runs for each entry of steady state peers only.

6.6.2. Adaptation Performance for Different Peer Class

Since not all peers are capable of adaptation, peers are classified into 3-tiers, which can do adaptation and streaming - others are assumed as not capable of performing any adaptation and considered as Tier-4. On average, for each adaptation, the CPU cycle consumed varies from 50% - 63%. To illustrate the idea of building the adaptation profile by the tree-controller, in Table 6, real-time spatiotemporal adaptation timing for different CPU specifications has been shown. From Table 6, we can see that the adaptation time is quite acceptable considering the fact that processing is being done in the compressed domain and in a live fashion. Here, adaptation time includes the adaptation of both the gBSD and the video.

TABLE 6. GBSD-BASED SPATIOTEMPORAL ADAPTATION PERFORMANCE FOR DIFFERENT PEER CLASS

Peer Class	CPU Specification	Time (Millisecond)
Tier-1	Pentium IV, 3.0 Ghz, 1GB RAM or Higher	52 – 135 Milliseconds
Tier-2	AMD Athlon XP, 1.4 Ghz, 512MB RAM or Equivalent	128 – 300 Milliseconds
Tier-3	Pentium III, 700 Mhz, 256 MB RAM or Equivalent	210 – 415 Milliseconds
Tier-4	Handheld Devices (ARM926T OMAP850 or Equivalent)	Not Applicable
Original encoded video: 1 Frameset - 30 frames, CIF		

6.6.3. Impact of Wealthier Nodes in the Overlay

From the simulation results for overlay construction, it is observed that the overall performance of the design is quite dependent on the participating nodes. The density of Tier-1 and Tier-2 parents plays an important role, as can be seen from Table 7, which drives the success of a particular overlay. Therefore, unless otherwise mentioned, during this simulation, presence of Tier-1, Tier-2, Tier-3 and Tier-4 peers varies from 10% to 15%, 20% to 30%, 20% to 30% and 20% to 25%, respectively.

TABLE 7. IMPACT OF WEALTHIER PEERS' AVAILABILITY ON SUCCESSFUL OVERLAY GENERATION

Tier-1 (%)	Tier-2 (%)	Tier-3 (%)	Tier-4 (%)	Success Rate (%)
10	0	0	90	16-20
20	0	0	80	22-30
30	0	0	70	40-46
40	0	0	60	57-60
50	0	0	50	81-85
10	40	0	50	60-68
10	30	10	50	55-60
10	20	30	50	50-55

6.6.4. Resource availability and QoE

From Figure 27 and Figure 28, we can see that both BSI and CSI gradually increase as the system receives more peers. As a result, multiple parent approach is able to serve a larger number of peers due to virtual resource concatenation. For the SDI (Figure 29), the extreme case is considered, in which every parent of a node may depart ungracefully. Usually, a receiver peer does not starve entirely unless all of its parents have left the overlay. From Figure 29, we can see that receiving peers that have multiple parents suffer less annoyance in terms of parent departure than those that have single or dual parents.

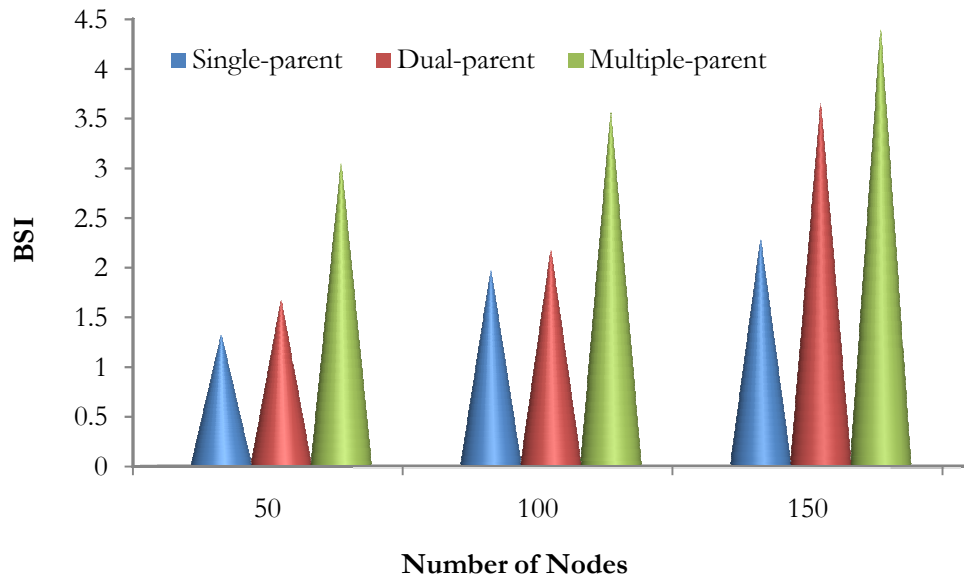


FIGURE 27. BSI FOR DIFFERENT NUMBER OF PARENTS

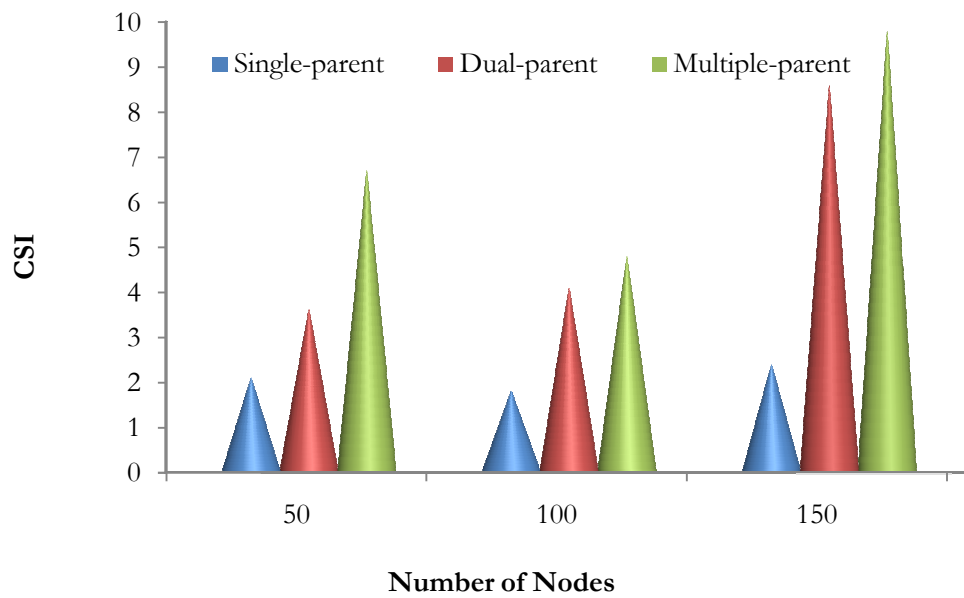
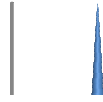


FIGURE 28. CSI FOR DIFFERENT NUMBER OF PARENTS



FIGUR

6.6.5. New Node Joining Suc

In Figure 30, Figure 31 parent approach and the multi node joining success rate. In b every 15-time units in the secc Figure 30, it can be observed th after considering the peer hete higher success rate even thoug if batching is enabled, the contr incoming peers.

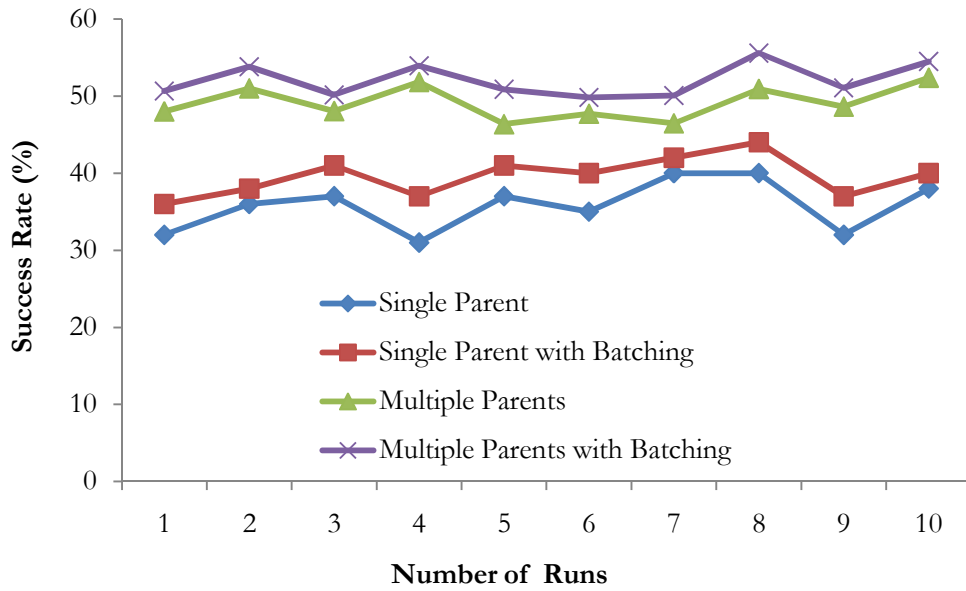


FIGURE 30. NEW NODE JOINING SUCCESS RATE

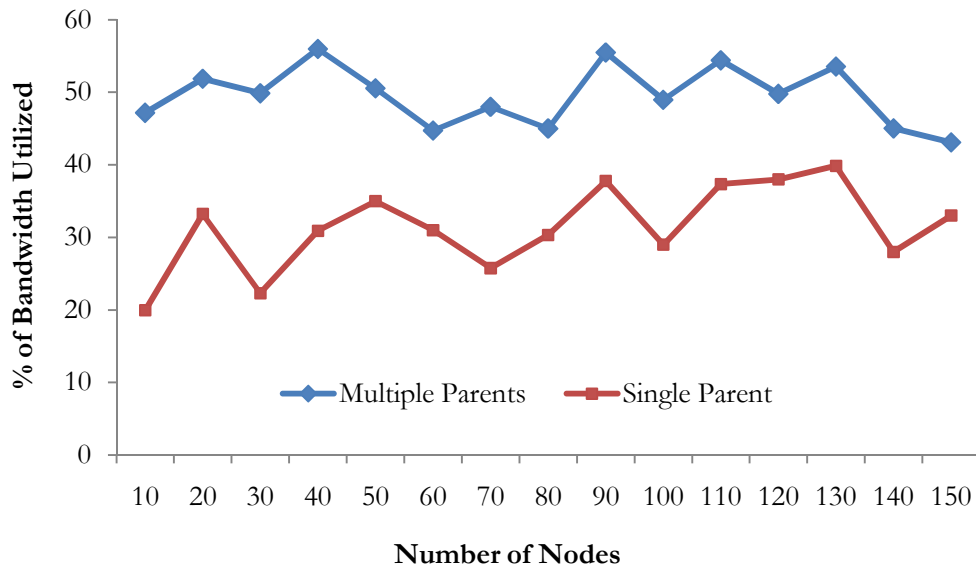


FIGURE 31. PERCENTAGE OF TOTAL BANDWIDTH USAGE

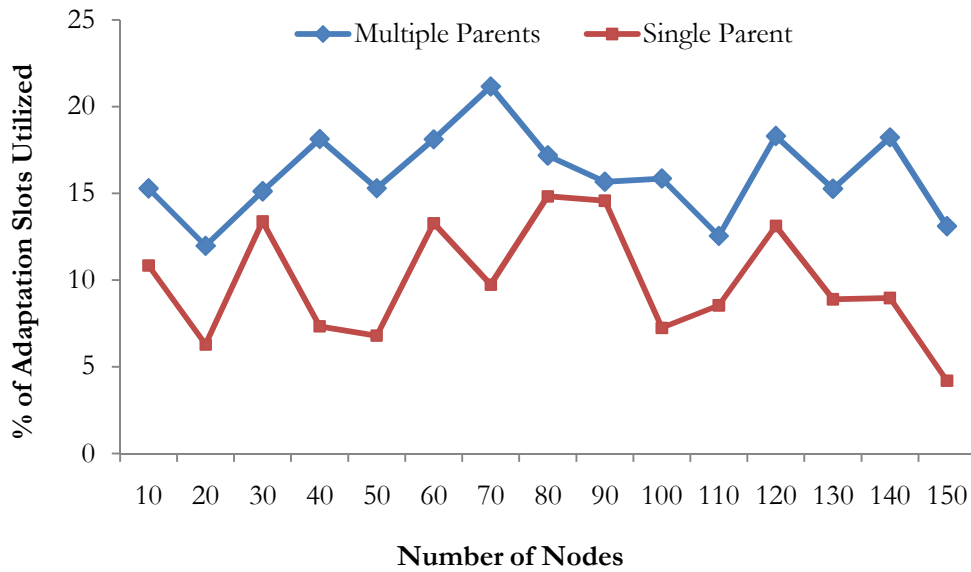


FIGURE 32. PERCENTAGE OF TOTAL ADAPTATION SLOTS USAGE

6.6.6. Resource Utilization

In Figure 31 and Figure 32, the percentage of total system bandwidth and adaptation time slots utilized in the overlays is shown, respectively. The total bandwidth utilization (Figure 31) is higher in the multiple parent approach, which is quite pertinent to the previous result that the success rate is also higher for the multiple parent option.

Now, compared to the bandwidth usage, from Figure 32 we see that adaptation slots utilization is comparatively low. This is due to the fact that there can be an ordinary peer requesting a low quality video, which can then serve other peers requesting the same video quality without any further adaptation need. Moreover, if a node is serving a peer with some specific spatiotemporal rate, then it can serve other incoming peers requesting the same quality video if the parent has enough upload bandwidth.

6.6.7. Internal Node Joining Success Rate

A regular peer disconnection (graceful or ungraceful), peer mobility, and the presence of handheld devices affect the system performance due to the high frequency of node joining and departing the network (i.e. Churn rate). Obviously, the churn rate of handheld devices is much higher than that of fixed nodes in the wired networks. Since Tier-1, Tier-2, and Tier-3 peers are responsible for adaptation and streaming services, and Tier-4 peers are mostly handheld devices who join the overlays as free riders, nodes joining and leaving the network from Tier-4 does not really hamper the topological properties. Figure 33 shows the effect of churn, which is due to the peers that get disconnected and subsequently request a reconnection. The internal node joining success rate is quite high because higher priority is given to servicing disconnected peers than to new connection requests. The effect of churn is measured for the steady-state peers only.

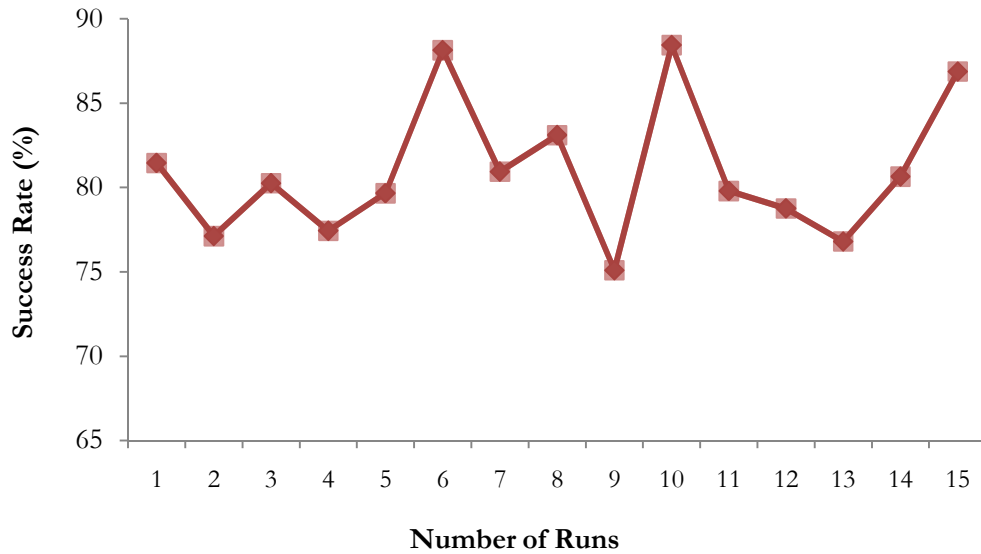


FIGURE 33. EFFECT OF CHURN: INTERNAL NODE JOINING SUCCESS RATE

6.6.8. Effectiveness of Minimum Contribute Rate

Figure 34 shows the change in available resources and node joining success rate when the minimum contribution rate is imposed. Overall, resource availability increases consistently, indicating the efficiency of the constraint to alleviate resource scarcity in the adaptive streaming system. However, node joining success rate may not increase with the same pace. This experience can be explained as follows - as the contribution rate increases, the peers become more altruistic, however, due to the peer churn or limited number of active peers, the resource sharing may not be always possible.

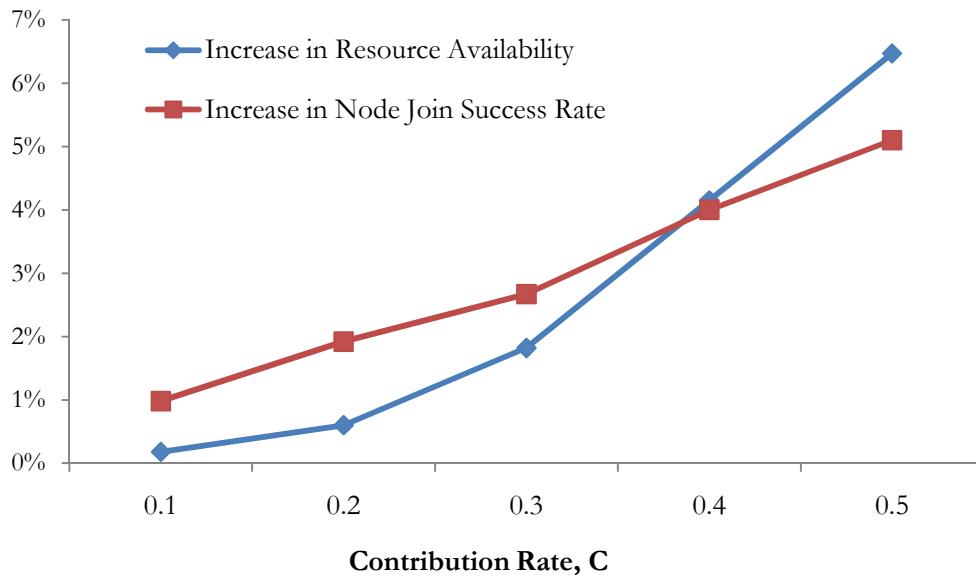


FIGURE 34. IMPACT OF THE CONTRIBUTION RATE

6.6.9. Effect of the Dynamic Quality Adjustment Scheme

The dynamic quality adjustment scheme benefits all the nodes that request adapted videos. Assuming a minimum of 10 framesets buffer for each node and for setting the retry time, $T=2$ unit time, the simulation is run ten times. Simulation results show that the quality adjustment scheme benefits on average 27% to 35% newly joined nodes where reduction in the

initial wait time ranges from 15% to 22%. On the other hand, average-waiting time to switch to the requested quality Q_i varies from 3 to 10 unit times. Longer switching times are mainly due to the peer churn and changing between different videos.

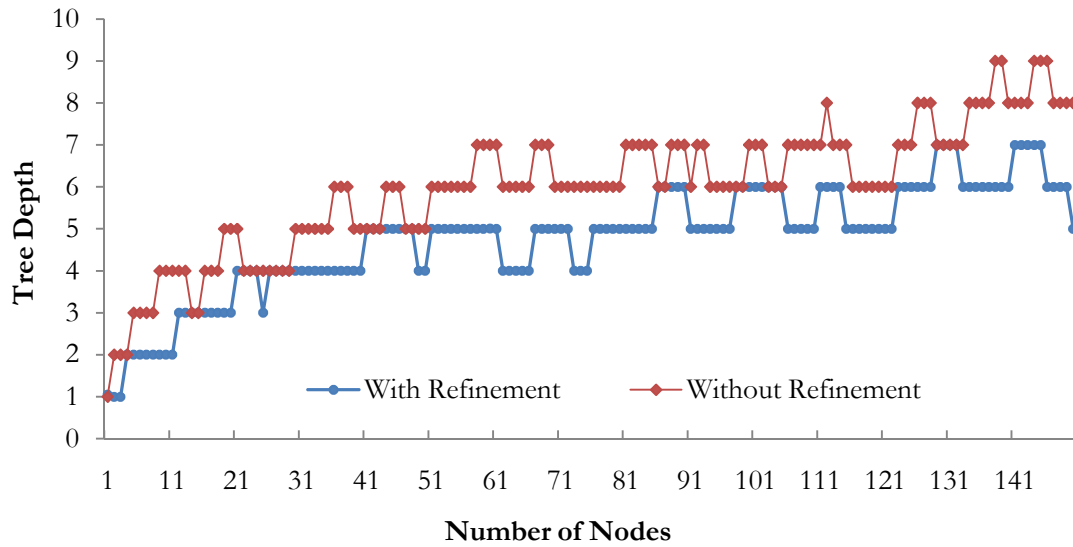


Figure 35. Tree refinement performance

6.6.10. Performance of Tree Refinement

The tree refinement operation applied by the tree-controller also enhances the system performance by minimizing tree height through the shifting of resourceful peers at the top of the tree. Figure 35 compares tree depth before and after refinement operation against the number of peers. In Figure 35, we can see that before refinement, the highest tree depth is 9, whereas, after refinement, the highest tree depth is 7. It was observed that the cumulative delay of some peers is more than the threshold (i.e. 300ms) because of the fact that in some cases the tree-controller may not be able to satisfy the refinement conditions. In such cases, no replacement operation will take place. From the simulation results, the average peer stretch is also computed, which is 4.5 and 6.5 for the single parent approach and the multiple parent approach, respectively. The peer

stretch is actually quite promising, especially when we need to consider the node joining constraint, upload bandwidth, and CPU requirements for adaptation operations.

CHAPTER 7

FINDING AN OPTIMUM OVERLAY

In the proposed multi-parent adaptive video streaming system, there exists a centralized node, denoted as a tree-controller, which gathers all the information from peers and decides on the overlay. Therefore, peers send the information regarding the amount of their upload and download bandwidth as well as the amount of processing power that they can share (or they may require) to the controller at joining time. At any given instant of time, a participating peer in a video overlay will receive video chunks from the source or from other peers (or both). This same peer will eventually participate in redistributing zero or more of the received chunks to other peers. Now, depending on the resource availability and given environmental constraints, we need to find an optimum overlay in order to evaluate the performance of the proposed design. In this chapter, an analytical model based on Integer Linear Programming is presented to evaluate the performance of the adaptive video streaming. The global knowledge of the overlay is collected from the tree-controller to calculate the optimum values.

For ease of understanding, in Section 7.1, the basic model is presented. This basic model is utilized to find the optimum overlay maximizing the available peer resources where each receiving node is being served by a maximum of one parent. In Section 7.2, an enhanced model is presented. In the enhanced model, service fairness, minimum peer contribution, and multiple-parent based video distribution constraints are integrated.

7.1. Problem Formulation

In a video distribution network, where the total number of peers is n for a video, cross connections for that video can be defined in the form of a matrix P , where $P := (p_{ij})_{n \times n}$. Here, each of the entries in P is a decision variable to indicate if there is a connection from node i to

node j , meaning that i directly serves j with no other peer nodes in-between. The value of each of these variables is a Boolean value where one indicates the existence of a connection between two arbitrary nodes i and j as demonstrated in Figure 36. In this matrix, the diagonal entries represent the state of connection between a node and itself. Therefore, we can remove each of these entries from the connection matrix, which will result in a total of $(n-1)^2$ number of connection variables. The problem of optimum overlay generation can then be redefined as finding the best values for p_{ij} variables so that the overlay can take the best advantage from available resources to serve the peers.

$$\begin{pmatrix} 0 & P_{1,2} & P_{1,3} & \cdots & P_{1,n} \\ P_{2,1} & 0 & P_{2,3} & \cdots & P_{2,n} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ P_{n,1} & P_{n,2} & P_{n,3} & \cdots & 0 \end{pmatrix} \rightarrow \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 1 & 0 & 0 & \cdots & 1 \\ \vdots & \vdots & & \cdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{pmatrix}$$

FIGURE 36. CONNECTION MATRIX, P

The tree-controller checks the availability of a specific amount of required download bandwidth and processing power in the system shared by the existing node to serve a new peer request. In order to simplify the problem, we assume that all the incoming peers are served by a maximum of one parent. Since, each video is decomposed into cb number of framesets, all the video framesets will be served by the designated parent. Consequently, a new node-joining request is either fully served or denied service. However, if denied, the new peer may reduce its demand and place a new service request. In Table 8, all the notations used in this section are summarized.

Now, for a certain node, i , the service that it provides its descendants can be defined as the sum of percentage of bandwidth and processing power given over all nodes j that are serviced by i . Therefore, it can be described as below:

$$S_i = \sum_j p_{ij} (db_j + cp_j) \quad \text{where } j=1\dots n, \quad i \neq j \quad (1)$$

Hence, the total overlay service provided by the participating nodes can be found as the summation of all single node services as defined in Equation 2.

$$S = \sum S_i = \sum_i \sum_j p_{ij} (db_j + cp_j) \quad \text{where } i=1\dots n, \quad j=1\dots n, \quad i \neq j \quad (2)$$

TABLE 8. SUMMARY OF NOTATIONS

n	<i>Total number of peers in the P2P streaming network</i>
cb	<i>Total number of framesets in a video</i>
ub_i	<i>i^{th} peer's total upload bandwidth, $i \in \{1\dots n\}$</i>
db_i	<i>i^{th} peer's required (normalized) download bandwidth for a video stream, $i \in \{1\dots n\}$</i>
ap_i	<i>i^{th} peer's available processing power, $i \in \{1\dots n\}$</i>
cp_i	<i>i^{th} peer's consuming (normalized) processing power, $i \in \{1\dots n\}$</i>
<i>In connection matrix, P, $p_{ij} = 1$ iff node j is served by node i otherwise $p_{ij} = 0$. $i \in \{1\dots n\}, j \in \{1\dots n\}, i \neq j$</i>	

Now, we need to find an optimum overlay in a way to maximize the total service provided in the overlay as previously described in Equation 2. The problem can be further redefined as a combinational optimization problem of finding the values of P matrix elements in order to maximize the total service as the cost function. The problem format is defined in Equation 3-7.

Maximize

$$\sum_i \sum_j p_{ij} (db_j + cp_j) \text{ where } i=1\dots n, j=1\dots n \text{ and } i \neq j \quad (3)$$

Subject to

$$\sum_i p_{ij} \leq 1 \quad (4)$$

$$\sum_j p_{ij} db_j \leq ub_i \quad (5)$$

$$\sum_j p_{ij} cp_j \leq ap_i \quad (6)$$

$$p_{ij} \in \{0, 1\}^n \quad (7)$$

Equation 4 above describes the fact that a requesting peer j can only be serviced by a single node i . Equation 5 is describing the constraint that the total bandwidth provided by peer i cannot exceed its total upload bandwidth, and Equation 6 defines that the total serviced CPU power of peer i cannot exceed its total shared processing power. The problem defined in Equation 3-7 is an Integer Linear Programming problem. Taking this as the base model, a mathematical case study is conducted to compare the effectiveness of the heuristic approach. Now, let us rewrite the above ILP problem in a general form as described in Section 2.4:

$$L = \{P_{ij} \mid i \in \{1 \dots n\}, j \in \{1 \dots n\}, i \neq j\}$$

where, the vector L includes all variables of the optimization problem

$$C = \{c_j \mid c_j = db_{j \bmod n} + cp_{j \bmod n}, j \in \{1 \dots n(n-1)\}\}$$

$$A_{(3n \times n(n-1))} = \{a_{ij} \mid i \in \{1 \dots 3n\}, j \in \{n \dots (n-1)\}\}$$

where

$$a_{ij} = \begin{cases} db_{j \bmod n} & j = n \dots (n-1), i = 1 \dots n \\ cp_{j \bmod n} & j = n \dots (n-1), i = (n+1) \dots 2n \\ 1 & j = n \dots (n-1), i = (2n+1) \dots 3n \end{cases}$$

$$b = \{b_i \mid i \in \{1 \dots 3n\}\}$$

where

$$b_i = \begin{cases} ub_i & i = 1 \dots n \\ ap_i & i = n+1 \dots 2n \\ 1 & i = 2n+1 \dots 3n \end{cases}$$

7.2. Revised ILP Model

Considering the ILP model formulated in the previous section as a base model, in this section, the base model is extended for the multiple parent option. Service fairness and minimum resource contribution constraints are also incorporated in this revised model.

In the multiple parent approach, a peer is serviced by multiple numbers of parents for a particular video; however, each frameset is delivered by only one parent. Whereas in the single parent approach, a receiving peer is solely dependent on one parent to receive the entire video. Since the H.264/AVC gives us non-layer video, therefore, in the single parent approach, a single connection matrix represents the optimal streaming overlay for the entire video. Consequently, for the multi-parent approach, a separate tree is formed for each frameset. Therefore, if we

define a connection matrix for each frameset, then the optimal result for all connection matrixes will cumulatively give the optimal result for the entire overlay for that particular video stream.

Now, let \mathcal{V} be the set of peers in the overlay, and the streaming source be denoted as S_t , where $S_t \in \mathcal{V}$. Transmission delay, $D_{s_t, j}$ refers to the sum of total delay from the stream-source to a node j involving the intermediary nodes along the delivery path. The transmission delay should be less than the delay threshold, DT_t . In the optimization problem as defined in Equation 8-19 below, the optimization goal is set to maximize the service ration, S , for the entire overlay where the value of S varies from 0 to 1. S determines the ratio of demand fulfillment for the peers.

Equation 9 below covers the facts that an incoming peer j will be served by a single parent i for a particular clip of the video. Equation 10 and Equation 11, respectively, defines that total bandwidth provided by peer i cannot exceed its total upload bandwidth, and that the total serviced CPU power of peer i cannot exceed its total shared processing power while serving all the children connected to it.

Equation 12 imposes the constraint that each peer is forced to contribute a certain amount of resource to the network. Equation 13 ensures that there is no cycle in the overlay for each chunk. Here, PV is defined as a set of vertices, and PE as a set consisting of edges connecting the elements in PV . Since $P_{ij}=1$ represents a link between i and j (i.e. an edge), we can also consider P_{ij} as a decision variable; therefore, the constraint in Equation 13 recursively guarantees that the number of included edges is equal to or less than the number of included vertices minus one. Equation 14 implies that there is no flow to the stream-starter and Equation 15 indicates that no node transmits to itself. Equation 16 ensures that the delay threshold is maintained along the delivery path from the stream-source to a receiver node.

Maximize

$$\text{Service Ratio, } S \quad (8)$$

Subject to

$$\sum_i p^k_{ij} \leq 1 \quad (9)$$

$$\sum_j p^k_{ij} \cdot s \cdot db_j \leq ub_i \quad (10)$$

$$\sum_j p^k_{ij} \cdot cp_j \leq ap_i \quad (11)$$

$$\sum_j p^k_{ij} (cp_j + db_j) \geq RG_i \quad (12)$$

$$\sum_j p^k_{ij} \leq |PE^k| - 1, \text{ where } (i, j \in PV^k), PE^k \neq \phi \quad (13)$$

$$p^k_{i \text{ st}} = 0 \quad (14)$$

$$p^k_{ii} = 0 \text{ if } i=j \quad (15)$$

$$p^k_{ij} \cdot D_{st,j} \leq DT_1 \quad (16)$$

$$p^k_{ij} \in \{0, 1\}^v \quad (17)$$

$$\forall k, 1 \leq k \leq ch \quad (18)$$

$$0 \leq S \leq 1 \quad (19)$$

7.3. Finding the Solution

According to the author in [92], optimization of a linear Pseudo-Boolean term CL , subject to a set of linear Pseudo-Boolean inequalities E , can be done by solving a sequence of Pseudo-Boolean satisfiability problems. Here we consider the problem of maximizing CL , thus, the goal is to find a solution α of E such that $\alpha(CL) \geq \alpha'(CL)$ for all solutions α' of E .

Such an assignment can be found by solving a sequence of satisfiability problems of the form $E_i = E \cup \{CL \geq max_i\}$ where only max_i differs from problem to problem. Suppose max_0 is in such a state that $CL \geq max_0$ is a tautology. Solving E_0 yields a solution α_0 of E_0 and eventually of E . A lower bound of the optimum then is $\alpha_0(CL)$. We define $max_{i+1} := max_i + 1$. Afterwards, if E_i is satisfiable and E_{i+1} is unsatisfiable, then $\alpha_i(CL)$ is the desired optimum.

The problems defined in Equation 3-7 and Equation 8-19 is in the form of a 0-1 ILP problem. As mentioned in Section 2.4, 0-1 ILP is the special case of Integer Programming where variables are required to be 0 or 1. This problem is categorized as NP-hard, therefore, there is no efficient ILP algorithm [93]. However, the above formulations of the overlay generation problem can be solved by the Simplex method [21]. Generally, the Simplex method is efficient, but it has a worst-case exponential running time [94]. There are also some worst-case polynomial time methods (e.g., Karmarkar's Algorithm⁸) for solving general LP optimization problems, however, these methods are quite complex to implement and suffer from a very large run-times. According to the authors in [95], in many cases, worst-case polynomial time methods surpass the average running time of the Simplex method.

To solve this problem, instead of implementing a new solver, CPLEX⁹ and MINOS¹⁰ solvers are used. The above ILP problems are solved using AMPL¹¹ modeling language and both CPLEX and MINOS solvers. During the simulation, the maximum number of simultaneous nodes is set to 150 for an overlay because as the number of nodes increases in a streaming session, the number of constraints in the LP problem increases and the efficiency of the Simplex algorithm begin to degrade. The node connection matrix found from the CPLEX (for the base model) and MINOS (for the revised model) solvers are analyzed to compute the total resource utilization in the optimal overlay.

7.4. Validation of the Heuristic Approach

Lack of similar designs and/or concepts in the literature is the primary reason why the performance of the proposed adaptive video streaming system cannot be compared with any other benchmark. To address this issue, we formulated the Integer Linear Programming problem

⁸ [online] http://en.wikipedia.org/wiki/Karmarkar's_algorithm

⁹ [online] www.cplex.com

¹⁰ [online] http://sbsi-sol-optimize.com/asp/sol_product_minos.htm

¹¹ [online] www.ampl.com

based model in order to find the optimal overlay and to compare the results found from optimal overlay in terms of resource utilization with the heuristic approach. In Figure 37 and Figure 38, a comparison of the single parent approach with the theoretical optimal overlay is presented. In Table 9, validation of the multi-parent approach and effectiveness of the service fairness constraint is elaborated.

From Table 9, we can see that the density of Tier-1 peers plays an important role to build a successful overlay. For example, when Tier-1 is 5%, the maximum node-join success rate achieved is 83% by offering 60% (i.e. Service Ratio, $S=0.6$) of the requested video quality to all peers; whereas, when Tier-1 is 20%, we could achieve an 84% joining success rate by offering a full rate of service (i.e. Service Ratio, $S=1.0$). Additionally, the optimum overlay was able to accommodate all the nodes offering 90% of the requested video quality to all peers. In general, as we decrease S , it is easier to accommodate more peers and hence, increases the utilization of the system resources. For the heuristic approach, bandwidth and CPU utilizations are consistently over 67% and 40%, respectively. If we consider peer mobility and peer disconnections, which is typical in any P2P system, then the overall resource utilization indicates the efficiency of the heuristic approach. From Figure 37, Figure 38, and Table 9, we can see that the performance of the heuristic approach closely follows the resource utilization boundary of the optimal overlay in terms of both bandwidth and CPU utilization.

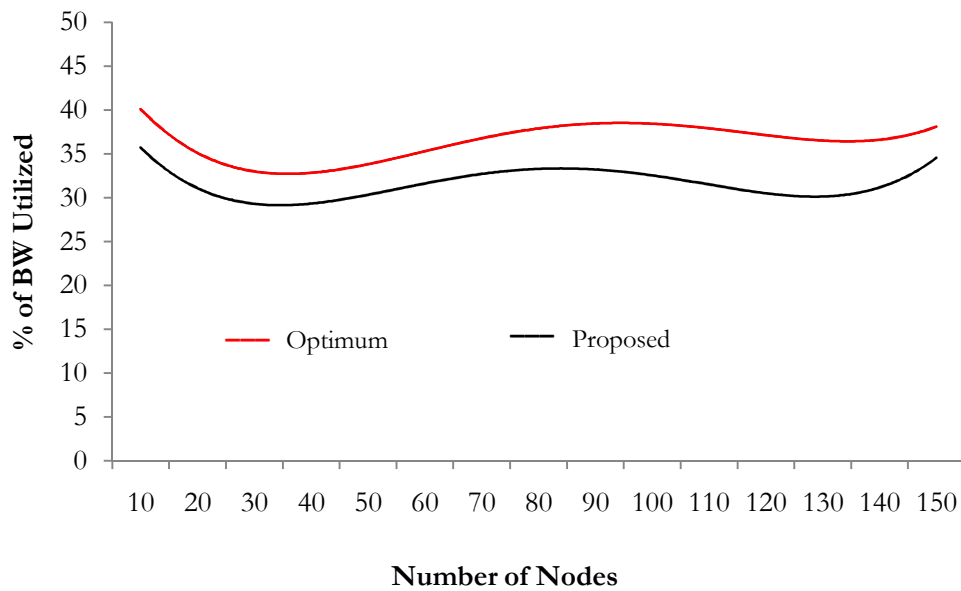


FIGURE 37. BANDWIDTH UTILIZATION IN SINGLE PARENT APPROACH - OPTIMUM VS. PROPOSED

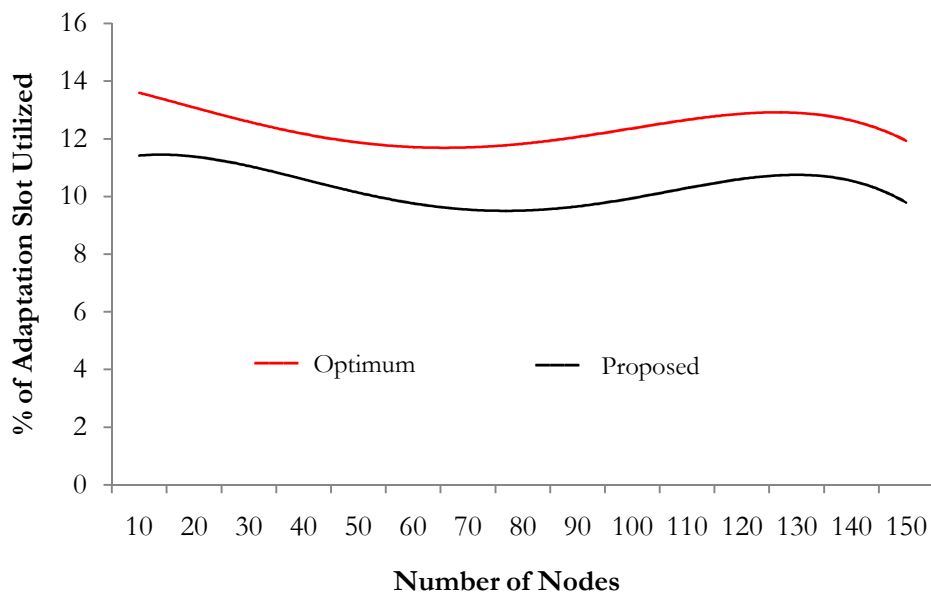


FIGURE 38. ADAPTATION SLOT UTILIZATION IN SINGLE PARENT APPROACH - OPTIMUM VS. PROPOSED

TABLE 9. VALIDATION OF THE MULTI-PARENT ADAPTIVE STREAMING SYSTEM: OPTIMUM VS. HEURISTIC

Peer Classification	Service Ratio, S	Optimum		Heuristic		
		Bandwidth Utilization	CPU Utilization	Bandwidth Utilization	CPU Utilization	Node Join Success
Tier-1: 5% Tier-2: 35% Tier-3: 35% Tier-4: 25%	0.6	89.0%	68.0%	80.6%	58.1%	83%
	0.7	84.6%	63.2%	76.1%	49.0%	81%
	0.8	82.7%	58.8%	72.1%	43.6%	76%
	0.9	79.5%	53.3%	67.3%	40.7%	73%
Tier-1: 10% Tier-2: 30% Tier-3: 35% Tier-4: 25%	0.8	86.7%	62.5%	81.0%	49.2%	84%
	0.9	84.2%	58.1%	76.7%	43.4%	81%
	1.0	80.1%	55.6%	72.9%	40.2%	76%
Tier-1: 20% Tier-2: 30% Tier-3: 30% Tier-4: 20%	0.9	87.2%	52.7%	81.3%	44.3%	88%
	1.0	84.6%	49.1%	78.1%	40.1%	84%

CHAPTER 8

CONCLUSION

8.1. Federated Video Processing and Online Streaming

Video adaptation is the process of transforming an input video to an output video and manipulating bitstream entities and segments in order to meet diverse environmental constraints as well as user preferences while optimizing the overall utility of the video. The main contribution of this research is the incorporation of a video adaptation scheme into the classical P2P streaming system. In this design, participating peers act as the stream-source and adaptation engine, while also performing the streaming tasks to serve heterogeneous devices including small handhelds. Intuitively, there is no need for streaming and/or adaptation servers. Metadata support for video adaptation seems to be a promising solution because it allows adaptation of video contents in a format-independent way.

The proposed federated video processing and online streaming concept is supported by a lightweight design where the core operations are simple to implement and easy to deploy as a client-server application. In this design, a peer is not allowed to communicate with all the existing peers, except the tree-controller, in order to maintain the scalability of the overlay. Moreover, instead of using fixed out-degree for each node, peer out-degree is calculated dynamically based on the sending throughput, which gives a better indication of resource availability and makes the architecture realistic. In order to balance between efficient resource usage and service fairness, constraints such as a minimum resource contribution requirement is added to the design. Moreover, the ILP-based mathematical model helps to judge the efficiency of the system at any given time. The benefit of the multiple parent approach and splitting video streams into framesets is that the overall resiliency of video overlays is improved since a node will not completely starve by the failure of a parent. As losses are often temporarily correlated along each

path, splitting the video framesets between different independent routes can be seen as a way to protect bitstream from consecutive losses. Although a scalable or a layer coded video could be an option (e.g. [99]) instead of real-time video adaptation, it should be noted that we may not achieve a fine grained adaptation performance in layer encoded videos since the more the number of layers, the less the coding efficiency. Additionally, stream-starter will require high CPU load in order to encode the scalable videos. On the other hand, benefit of H.264/AVC coding is that there is no layer dependency and all the overlays can have an equal number of peer contributions. Otherwise, overlays for base layers have to have a higher number of peers than that of higher layers. From the simulation results, it is also quite evident that the difference in the performance of the proposed design and the optimal solution is tightly bounded.

8.2. General Discussion

Until now, P2P video streaming concept has been limited to sharing only bandwidth and little has been done to investigate the utilization of idle CPU resources of the participating peers. Therefore, adding the functionality of content adaptation to ordinary video streaming applications is a timely initiative to open up the race for second-generation video streaming systems. From this research, it is evident that if the participants in a community network agree to not only share their bandwidth but also their computing resources, then heterogeneous devices can be accommodated ensuring adequate resource utilization with respect to resilience to peer dynamics. Now the question is - who will provide the tree-controller? Perhaps this could be a commercial service provider. Their benefit is that they could earn revenues by showing advertisements in the application window as we usually see in the free VOIP services. Notably, tree-controller only coordinates the tree formation centrally, whereas, streaming and adaptation tasks remain distributed among the participating peers. Moreover, the reason behind choosing a tree-based approach is that there will be minimal protocol overhead for tree maintenance and coordination at each node. Otherwise, if mesh-based approach is applied then it results in partitioning and slow recovery when the peers do channel surfing.

The applicability of the proposed adaptive video architecture is not only limited to online video streaming/distribution, but it also has its scope in a range of media-driven applications such as video surveillance. For example, recent international incidents precipitating a focus on domestic security issues, illustrate the need for distributed camera network architecture with decentralized lookup and video adaptation for heterogeneous nodes. In emergencies, such as the unfortunate Mumbai incident¹², emergency services' response is often thwarted due to imprecise situational knowledge. If security cameras or sensor nodes with wireless transmission facilities are deployed in the environment, these may act as access points for an emergency official's handheld device. When the official comes within connection range of these cameras, the device connects to the camera overlay allowing video streams to be requested from a location or location range. This helps the officials to see immediate and remote corridors, rooms, and corners without the need of a functioning centralized architecture. Once a link between the node and the receiver is established, network and device heterogeneity must be taken into account for live video broadcasting or streaming. Sparked by the desire to secure human presence in high-risk areas, in [100] we proposed a distributed, scalable, and fault tolerant camera network architecture. The architecture is a robust platform for video surveillance systems, which has the ability to collect/capture video streams from cameras at a point-of-interest and deliver a consolidated version of the video to heterogeneous receivers. Besides, the proposed video processing schemes can also be deployed as a tool/plug-in for web-services. Surely, latency will increase due to the real-time video processing operations. Therefore, some may suggest using multi-layer encoding in order to accommodate heterogeneous nodes without any additional overhead of bandwidth or CPU power. Conversely, it is also true that not many devices are capable of decoding such streams; and as mentioned earlier, the encoding process may consume high CPU power at the stream-starter. However, the fact cannot be avoided that the practical impact of the proposed technique might be limited if the stakeholders are reluctant to adopt the MPEG-21 multimedia framework.

¹² BBC News, "Mumbai rocked by deadly attacks", November 2008. [Online]. Available: http://news.bbc.co.uk/2/hi/south_asia/7751160.stm

8.3. Future work

Online adaptation and adding up heterogeneous receivers to overlay streaming is a new concept. In the adaptive video streaming design presented in this thesis, it is assumed that the rendezvous point is the tree-controller and that it will not fail, which in practice might not hold true all the time. Therefore, keeping in mind the drawbacks of the centralized design, the possibility and feasibility of a decentralized adaptive video streaming system needs to be investigated. In this regard, focus needs to be given on the following two issues: i) Node joining delay due to parent discovery in the decentralized design, and ii) Impact of the control message flooding in case of node departure or new node joining.

Research in context awareness, especially in digital video contents, is also a relatively new field, and therefore, rational findings are scattered. Existing literature works mainly focus on context types such as identity, location, time, activity, etc. Therefore, the videography priorities to intuitively deduce and interpret region of interests in H.264 videos is another field of research potential. Finally, formulation of an automated adaptation decision-making mechanism for varying environment will help the ubiquitous computing concept come to fruition.

BIBLIOGRAPHY

- [1] comScore Press Release[Online]. <http://www.comscore.com/press/release.asp?press=1529>
- [2] E. Setton, P. Baccichet, and B. Girod. Peer-to-Peer Live multicast: A Video Perspective. In Proceedings of IEEE, vol. 96, no. 1, pp. 25-38, 2008.
- [3] A. Richard. YouTube: Overnight Success has Sparked a Backlash. [Online]. <http://www.telegraph.co.uk/news/uknews/2480280/YouTube-Overnight-success-has-sparked-a-backlash.html>
- [4] X. Jin, K.-L. Cheng, and S.-H. G. Chan. Scalable Island Multicast for Peer-to-Peer Streaming. In Advances in Multimedia, vol. 2007, issue 1, 2007.
- [5] X. Jin, K.-L. Cheng, and S.-H. G. Chan. Island Multicast: Combining IP Multicast With Overlay Data Distribution. In IEEE Transactions on Multimedia, vol. 11, no.5, pp. 1024-1036, 2009.
- [6] M. Hosseini, D.T. Ahmed, S. Shirmohammadi, and N.D. Georganas. A Survey of Application-Layer Multicast Protocols. In IEEE Communication Surveys and Tutorials, vol. 9, issue 3, pp. 58-74, 2007.
- [7] S. Banerjee, B. Bhattacharjee, and C. Kommareddy. Scalable Application Layer Multicast. In Proceedings of Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM), pp. 205-217, 2002.
- [8] D.A. Tran, K.A. Hua, and T.T. Do. A Peer-to-Peer Architecture for Media Streaming. In IEEE Journal on Selected Areas in Communications, vol. 2, no.1, pp. 121-133, 2004.
- [9] M. Castro, P. Druschel, A. Kermarrec, A. Nandi, A. Rowston, and A. Singh. Splitstream: High-bandwidth Multicast in Cooperative Environments. In Proceedings of ACM Symposium on Operating Systems Principles, pp. 298-313, 2003.
- [10] P. Baccichet, T. Schierl, T. Wiegand, and B. Girod. Low-delay Peer-to-Peer Streaming Using Scalable Video Coding. In Proceedings of Packet Video, pp. 173-181, 2007.
- [11] Y. Zhou, D. Chiu, L., and J. Lui. A Simple Model for Analyzing P2P Streaming Protocols. In Proceedings of International Conference on Network Protocols (ICNP), pp. 226-235, 2007.

- [12] ISO/IEC 21000-7:2004, Information Technology – Multimedia Framework – Part 7: Digital Item Adaptation.
- [13] A. Vetro and C. Timmerer. Digital Item Adaptation: Overview of Standardization and Research Activities. In IEEE Transactions on Multimedia, vol. 7, issue 3, 2005, pp. 418-426.
- [14] L. Rong and I. Burnett. Dynamic Multimedia Adaptation and Updating of Media Streams with MPEG-21. In Proceedings of IEEE Consumer Communications and Networking Conference (CCNC), pp. 436-441, 2004.
- [15] S. Wenger. H.264/AVC over IP. In IEEE Transactions on Circuits and Systems for Video Tech. vol.13, issue 7, pp. 645-656, 2003.
- [16] C. Gomila and P. Yin. New Features and Applications of the H.264 Video Coding Standard. In Proceedings of International Conference on Information Technology: Research and Education, pp. 6-10, 2003.
- [17] P. Barth. Logic-Based 0-1 Constraint Programming. ISBN: 0792396634, Kluwer Academic Publishers, 1996.
- [18] J. Clausen. Branch and Bound Algorithms - Principles and Examples. Department of Computer Science, University of Copenhagen, Denmark, 1999.
- [19] J. E. Mitchell. Integer Programming: Branch-and-Cut Algorithms. The Encyclopedia of Optimization, vol. II, pp. 519-525, 2001.
- [20] M. Davis and H. Putnam. A Computing Procedure for Quantification Theory. In Journal of ACM, vol. 7, issue 3, pp. 201-215, 1960.
- [21] G. Sierksma. Linear and Integer Programming: Theory and Practice. ISBN: 0824706730, CRC Press, 2002.
- [22] J. R. Smith, R. Mohan, and C. Li. Scalable Multimedia Delivery for Pervasive Computing. In Proceedings of ACM Multimedia, pp. 131-140, 1999.
- [23] A. Fox and E. A. Brewer. Reducing WWW Latency and Bandwidth Requirements by Real Timer Distillation. In Proceedings of WWW, vol. 28, issue 7-11, pp. 1445-1446, 1996.
- [24] MPEG-7 Overview v.9. International Standards Org./Int. Electrotech. Comm. (ISO/IEC) JTC 1, ISO/IEC JTC1/SC29/WG11N5525, 2003.
- [25] Exchange protocol based on HTTP extension framework [Online]. <http://www.w3.org/TR/NOTE-CCPPexchange>

- [26] TV-Anytime forum [Online]. <http://www.tv-anytime.org>
- [27] J. Xin, C.-W. Lin, and M.-T. Sun. Digital Video Transcoding. In Proceedings of IEEE, vol. 93, no. 1, pp. 84–97, 2005.
- [28] K. Taehyun and M.H. Ammar. Optimal Quality Adaptation for Scalable Encoded Video. IEEE Journal on Selected Areas in Communications, vol. 23, pp. 344-356, 2005.
- [29] M.A. Bonuccelli, F. Lonetti, and F. Martelli. Temporal Transcoding for Mobile Video Communication. In Proceedings of International Conference on Mobile and Ubiquitous Systems: Networking and Services, pp. 502-506, 2005.
- [30] F. D. Vito, T. Ozcelebi, R. Civanlar, A. M. Tekalp, and J. C. D. Martin. Rate Control For GOP-Level Rate Adaptation in H.264 Video Coding. In Proceedings of International Workshop on Very Low Bit-rate Video Coding, 2005.
- [31] I. Shin, Y. Lee, and H. Park. Motion Estimation for Frame-rate Reduction in H.264 Transcoding. In Proceedings of IEEE Workshop on Software Technologies for Future Embedded and Ubiquitous Systems, pp. 63-67, 2004.
- [32] P. Zhang, Y. Lu, Q. Huang, and W. Gao. Mode Mapping Method for H.264/AVC Spatial Downscaling Transcoding. In Proceedings of IEEE International Conference on Image Processing (ICIP), vol. 4, pp. 2781-2784, 2004.
- [33] J. Cock, S. Notebaert, and K. Vermeersch et al. Efficient Spatial Resolution Reduction Transcoding for H.264/AVC. In Proceedings of IEEE International Conference on Image Processing (ICIP), pp. 1208-1211, 2008.
- [34] Y. Wang, X. Fan, H. Li, Z. Liu, and M. Li. An Attention Based Spatial Adaptation Scheme for H.264 Videos on Mobiles. In Proceedings of Multi-Media Modeling Conference, 2006.
- [35] H.K. Arachchi, S. Dogan, H. Uzuner, and A.M. Kondoz. Utilizing Macroblock SKIP Mode Information to Accelerate Cropping of an H.264/AVC Encoded Video Sequence for User Centric Content Adaptation. In Proceedings of International Conference on Automated Production of Cross Media Content for Multi-Channel Distribution, pp. 3-6, 2007.
- [36] S. Devillers, C. Timmerer, J. Heuer, and H. Hellwagner. Bitstream syntax description-based adaptation in streaming and constrained environments. In IEEE Transactions on Multimedia, vol. 7, issue 3, pp. 463-470, 2005.
- [37] S.G. Deshpande. High Quality Video Streaming Using Content-aware Adaptive Frame Scheduling with Explicit Deadline Adjustment. In Proceedings of ACM Multimedia, pp. 777-780, 2008.

- [38] M.S. Hossain, A. Alamri, and A. El Saddik. QoS-Aware Service Selection for Multimedia Transcoding. In Proceedings of IEEE Instrumentation and Measurement Technology (I²MTC), issue 6, pp. 588-593, 2008.
- [39] T. Ahmed, A. Mehaoua, R. Boutaba, and Y. Iraqi. Adaptive Packet Video Streaming Over IP Networks: A Cross-layer Approach. In IEEE Journal on Selected Areas in Communications, vol. 23, issue 2, 2005.
- [40] B. Li and J. Liu. Multirate Video Multicast Over the Internet: An Overview. In IEEE Network, vol. 17, issue 1, pp. 24–29, 2003.
- [41] H. Hellwagner, R. Kuschnig, T. Stutz, and A. Uhl. Efficient in-network adaptation of encrypted H.264/SVC content. In Image Communication, vol. 24, issue 9, pp. 740–758, 2009.
- [42] I. Kofler, M. Prangl, R. Kuschnig, and H. Hellwagner. An H.264/SVC-based adaptation proxy on a WiFi router. In Proceedings of ACM International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV), pp. 63–68, 2008.
- [43] M. Noorkami and R. M. Mersereau. Towards Robust Compressed-Domain Video Watermarking for H.264. In SPIE Security, Steganography, and Watermarking of Multimedia Contents, vol. 6072, pp. 489-497, 2006.
- [44] G. Wu, Y. Wang, and W. Hsu. Robust Watermark Embedding/Detection Algorithm for H.264 Video. In Journal of Electronic Imaging, vol. 14, 2005.
- [45] G. Qiu, P. Marziliano, A.T.S. Ho, D. He, and Q. Sun. A Hybrid Watermarking Scheme for H.264/AVC Video. In Proceedings of International Conference on Pattern Recognition, vol. 4, pp. 865-868, 2004.
- [46] J. Zhang and A.T.S. Ho. Efficient Video Authentication for H.264/AVC. In Proceedings of International Conference on Innovative Computing, Information and Control (ICICIC), pp. 46-49, 2006.
- [47] D. Pröfrock, H. Richter, M. Schlaueg, and E. Müller. H.264/AVC Video Authentication Using Skipped Macroblocks for an Erasable Watermark. In Proceedings of SPIE Visual Communications and Image Processing, vol. 5960, pp. 1480-1489, 2005.
- [48] Y. Bodo, N. Laurent, and J.-L. Dugelay. A Scrambling Method Based on Disturbance of Motion Vector. In Proceedings of ACM Multimedia, pp. 89-90, 2002.

- [49] S. Lian, J. Sun, and Z. Wang. Perceptual cryptography on SPIHT compressed images or videos. In Proceedings of IEEE International Conference on Multimedia and Expo (ICME), vol. 3, pp. 2195-2198, 2004.
- [50] S. Lian, X. Wang, J. Sun, and Z. Wang. Perceptual cryptography on wavelet-transform encoded videos. In Proceedings of IEEE International Symposium on Intelligent Multimedia, Video and Speech Processing, pp. 57-60, 2004.
- [51] H. Cheng and X. Li. Partial Encryption of Compressed Images and Videos. In IEEE Transactions on Signal Processing, vol. 48, issue 8, pp. 2439-2451, 2000.
- [52] T. Lookabaugh. Selective Encryption, Information Theory and Compression. In Asilomar Conference on Signals, Systems and Computers, vol. 1, pp. 373-376, 2004.
- [53] T. Shi, B. King, and P. Salama. Selective Encryption for H.264/AVC Video. In Proceedings of the SPIE, vol. 6072, pp. 461-469, 2006.
- [54] R. Iqbal, S. Shirmohammadi, A. ElSaddik, and J. Zhao. Compressed Domain Video Processing for Adaptation, Encryption, and Authentication. In IEEE Multimedia, vol. 15, no. 2, pp. 38-50, 2008.
- [55] B. Li, S. Xie, Y. Qu, Y. Keung, C. Lin, J. Liu, and X. Zhang. Inside the New Coolstreaming: Principles, Measurements and Performance Implications. In Proceedings of IEEE INFOCOM, pp. 1031-1039, 2008.
- [56] P. Decuetos and K.W. Ross. Unified Framework for Optimal Video Streaming. In Proceedings of IEEE INFOCOM, vol. 3, pp. 1479-1489, 2004.
- [57] C. Jennigs and D. Bryan. P2P for Communications: Beyond File Sharing. In Business Communication Review, Feb. 2006.
- [58] N. Magharei, R. Rejaie, and Y. Guo. Mesh or Multiple-Tree: A Comparative Study of Live P2P Streaming Approaches. In Proceedings of IEEE INFOCOM, pp. 1424-1432, 2007.
- [59] N. Magharei and R. Rejaie. PRIME: Peer-to-Peer Receiver Driven Mesh-based Streaming. In Proceedings of IEEE INFOCOM, pp. 1415-1423, 2007.
- [60] X. Hei, C. Liang, J. Liang, Y. Liu, and K.W. Ross. A Measurement Study of a Large-Scale P2P IPTV System. In IEEE Transactions on Multimedia, vol. 9, issue 8, pp. 1672-1687, 2007.
- [61] A. Ali, A. Mathur, and H. Zhang. Measurement of Commercial Peer-to-Peer Live Video Streaming. In Proceedings of Recent Advances in Peer-to-Peer Streaming, 2006.

- [62] M. van der Schaar, D.S. Turaga, and R. Sood. Stochastic Optimization for Content Sharing in P2P Systems. In IEEE Transactions on Multimedia, vol. 10, issue 1, 2008, pp. 132-144.
- [63] V.N. Padmanabhan, H.J. Wang, P.A. Chou, and K. Sripanidkulchai. Distributing Streaming Media Content Using Cooperative Networking. In Proceedings of ACM International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV), pp. 177-186, 2002.
- [64] Y. Zhu, B. Li, and J. Guo. Multicast with Network Coding in Application-layer Overlay Networks. In IEEE Journal on Selected Areas Communication, no. 22, pp. 107-120, 2004.
- [65] M. Zhang, J. G. Luo, L. Zhao, and S.Q. Yang. A Peer-to-Peer Network for Live Media Streaming using a Push-Pull Approach. In Proceedings of ACM Multimedia, pp. 287-290, 2005.
- [66] A. Rodriguez, D. Kotic, and A. Vahdat. Scalability in Adaptive Multi-metric Overlays. In Proceedings of International Conference on Distributed Computing Systems, pp. 112-121, 2004.
- [67] G. Exarchakos and N. Antonopoulos. Resource Sharing Architecture for Cooperative Heterogeneous P2P Overlays. In Journal of Networks and Systems Management, vol. 15, pp. 311-334, 2007.
- [68] C. Wu, B. Li, and S. Zhao. Exploring Large-Scale Peer-to-Peer Live Streaming Topologies. In ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP), vol. 4, issue 3, 2008.
- [69] G. Tan, S. A. Jarvis, and D. P. Spooner. Improving the Fault Resilience of Overlay Multicast for Media Streaming. In IEEE Transactions on Parallel and Distributed Systems, vol. 18, issue 6, pp. 721-734, 2007.
- [70] E. Setton, P. Baccichet, and B. Girod. Peer-to-Peer Live Multicast: A Video Perspective. In Proceedings of IEEE, vol. 96, no. 1, 2008.
- [71] X. Tan and S. Datta. Building Multicast Trees for Multimedia Streaming in Heterogeneous P2P Networks. In Proceedings of Systems Communications, pp. 141-146, 2005.
- [72] N. Magharei and R. Rejaie. Adaptive Receiver-driven Streaming from Multiple Senders. In Multimedia Systems Journal, vol. 11, no. 6, pp. 1-18, 2006.
- [73] Y. Cui and K. Nahrstedt. Layered Peer-to-Peer Streaming. In Proceedings of ACM International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV), pp. 162-171, 2003.

- [74] T. Nguyen and A. Zakhor. Distributed Video Streaming Over Internet. In Proceedings of Multimedia Computing and Networking (MMCN), 2002.
- [75] R. Rejaie and A. Ortega. PALS: Peer-to-Peer Adaptive Layered Streaming. In Proceedings of ACM International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV), pp. 153-161, 2003.
- [76] X. Xu, Y. Wang, S.S. Panwar, and K.W. Ross. A Peer-to-Peer Video-on-Demand System using Multiple Description Coding and Server Diversity. In Proceedings of International Conference on Image Processing (ICIP), vol. 3, pp. 1759-1762, 2004.
- [77] B. Shen, W. Tan, and F. Huve. Dynamic Video Transcoding in Mobile Environments. In IEEE Multimedia, vol. 15, no. 1, pp. 42-51, 2008.
- [78] D. Liu, E. Setton, B. Shen, and S. Chen. PAT: Peer-Assisted Transcoding for Overlay Streaming to Heterogeneous Devices. In Proceedings of ACM International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV), 2007.
- [79] A. Ouali, B. Jaumard, and G. Hebuterne. Trade-offs in Peer Delay Minimization for Video Streaming in P2P Systems. In Proceedings of IEEE International Symposium on Cluster Computing and the Grid (CCGRID), pp. 615-620, 2008.
- [80] V. Venkataraman, P. Francis, and J. Calandrino. Chunkyspread: Heterogeneous Unstructured Tree-based Peer-to-Peer Multicast. In Proceedings of IEEE International Conference on Network Protocols (ICNP), pp. 2-11, 2006.
- [81] M. Piatek, T. Isdal, T. Anderson, A. Krishnamurthy, and A. Venkataramani. Do Incentives Build Robustness in BitTorrent?. In NSDI, Cambridge, April 2007.
- [82] M. Sirivianos, J.H. Park, R. Chen, and X. Yang. Free-riding in BitTorrent Networks with the Large View Exploit. In Proceedings of IPTPS, 2007.
- [83] D. Bertsimas and J. Tsitsiklis. Introduction to Linear Optimization. Athena Scientific Press, 1997.
- [84] C.A.S. Oliveira and P.M. Pardalos. A Survey of Combinatorial Optimization Problems in Multicast Routing. In Computers and Operations Research, vol. 32, no. 8, 2005.
- [85] C. Hu, E. Hsiao-Kuang, and G. Chen. Bandwidth-satisfied Multicast Trees in MANETs. In Proceedings of International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), vol. 3, pp. 323-328, 2005.
- [86] S-F. Chang and A. Vetro. Video Adaptation: Concepts, Technologies, and Open Issues. In Proceedings of IEEE, vol. 93, no. 1, pp. 148-158, 2005.

- [87] H.264 Reference Software [Online]. http://ftp3.itu.ch/av-arch/jvt-site/reference_software/
- [88] XSLT [Online]. <http://www.w3.org/TR/xslt>
- [89] A. V. Aho, R. Sethi, and J. D. Ullman. Compilers: Principles, Techniques, and Tools. Addison-Wesley, pp. 434-438, 1986.
- [90] Y. Sung, M. Bishop, and S. Rao. Enabling Contribution Awareness in an Overlay Broadcasting System. In Proceedings of Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM), pp. 411-422, 2006.
- [91] Y. J. Liang and J. G. Apostolopoulos. Analysis of Packet Loss for Compressed Video: Effect of Burst Losses and Correlation Between Error Frames. In IEEE Transactions on Circuits and Systems for Video Technology, vol. 18, no.7, pp. 861-874, 2008.
- [92] P. Barth. A Davis-Putnam Based Enumeration Algorithm for Linear Pseudo-Boolean Optimization. Technical Report, MPI-I-95-2-003, Max-Planck-Institut für Informatik, 1995.
- [93] T. Cormen, C. Leiserson, R. Rivest, and C. Stein. Introduction to Algorithms. 2nd ed. MIT Press, 2001.
- [94] D. Goldfarb and M. Todd. Linear Programming. Handbook in Operations Research and Management Science, vol. 1, pp. 73-170, 1989.
- [95] R. Marler and J. Arora. Survey of Multi-objective Optimization Methods for Engineering. Structural and Multidisciplinary Optimization, vol. 26, no. 6, pp. 369-395, 2004.
- [96] CPLEX Tool [Online]. <http://www.ilog.com>
- [97] JXTA [Online]. <http://www.sun.com/software/jxta/>
- [98] K. Sripanidkulchai, A. Ganjam, B. Maggs, and H. Zhang. The Feasibility of Supporting Large-scale Live Streaming Applications with Dynamic Application Endpoints. In Proceedings of Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM), pp. 107-120, 2004.
- [99] K. Mokhtarian and M. Hefeeda. Analysis of Peer-assisted Video-on-Demand Systems with Scalable Video Streams. In Proceedings of ACM Multimedia Systems (MMSys), pp. 133-144, 2010.
- [100] R. Iqbal, S. Ratti, and S. Shirmohammadi. A Distributed Camera Network Architecture Supporting Video Adaptation. In Proceedings of ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC), pp. 1-7, 2009.