AN ABSTRACT OF THE THESIS OF

Richard A. Kramer for the degree of Master of Science in Electrical and Computer Engineering presented on April 10, 2017

Title: Optimization of Interactive Live Free Viewpoint Multiview Video Streaming Bandwidth

Abstract approved: ____

Thinh P. Nguyen

There are generally two types of multiview video:

1) 3D Multiview Video ("3D MVV" also called "2D plus delta" or "stereo" multi-view video): 3D MVV is widely deployed in cinemas and in the TV industry. 3D MVV typically entails capturing video of an object using two cameras with differing view angles to allow for depth perception.
2) Free Viewpoint Multiview Video TV ("FTV"): Free Viewpoint Multiview Video allows the user to freely pan horizontally based on an array of cameras.

While advances have occurred for 3D MVV, and to a lesser extent, stored FTV [Cheung2011], the commercialization and advancement of live Free Viewpoint Multiview Video has stalled. The failure of FTV is attributed to the lack of standardization and the high bandwidth requirements of the FTV content.

> *"Today, there are no known streaming services that provide MVV [Multi-View Video] content to home users … the fundamental reasons for this can be listed as: (i) lack of specifications for MVV, such as resolution and number of views, making it difficult to create universal content that is suitable for all multiview displays; (ii) heterogeneous bandwidth requirement of different multiview displays, making it infeasible to perform transmission over fixed bit-rate channels"* ([Dufaux2013] at pg. 201).

This thesis provides a solution to obtain a significant reduction in FTV broadcast and P2P bandwidth utilization as compared to current proposed methods, while maximizing the user experience for the given bandwidth.

Optimization of Interactive Live Free Viewpoint Multiview Video Streaming Bandwidth

by
Richard A. Kramer

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented April 10, 2017
Commencement June 2017

Master of Science thesis of Richard A. Kramer presented on April 10, 2017

APPROVED

Major Professor , representing Electrical and Computer Engineering

Director of the School of Electrical Engineering and Computer Science

Dean of the Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Richard A. Kramer, Author

ACKNOWLEDGEMENTS

The author expresses sincere appreciation to Dr. Thinh P. Nguyen. Dr. Nguyen has shown amazing dedication to his students to which I am honored to be one of. Dr. Nguyen has taken me under his wing at many levels for which I am grateful.

I would also like to thank my dear wife, Luda, who has been an inspiration to me. Her understanding and encouragement has driven me to new levels.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

Section Page

LIST OF FIGURES

LIST OF FIGURES (Continued)

LIST OF FIGURES (Continued)

Figure Page

LIST OF TABLES

Chapter 1: Introduction

1.1 State of the Industry

In 2009, the H.264/MPEG-4 standard incorporated Free Viewpoint Multiview Video TV ("FTV") video compression as part of ISO/IEC standard 14496-10, Version 11 [ISO/IEC standard 14496-10:2009]. The MPEG-4 standard refers to this video compression model as the "Multiview High Profile" which further utilizes B-frames (Bi-Directional-frames) and P-frames (Predictive-frames) to take advantage of camera to camera redundancies [Kramer2016]. *While standardization exists with FTV video compression, standardization does not exist related to providing a practical transport solution.*

The last apparent effort to drive standardization related to FTV *transport* appears to be a European initiative called "DIOMEDES" (DIstribution Of Multi-view Entertainment using content aware DElivery Systems). The DIOMEDES effort produced a system prototype and set of standards [DIOMEDES D2.3 2012][DIOMEDES D3.6 2011][DIOMEDES D4.4 2011][DIOMEDES D4.5 2011], whereas the effort seemingly ended in 2012 [DIOMEDES D2.3 2012] and little if anything has been published on *live* FTV since that time. While DIOMEDES delivered a set of general standards, they appear to be incomplete and not adopted by the industry as a whole. To understand the DIOMEDES standards' short comings, this thesis provides an overview of the previous state of the art for live FTV, focusing on where DIOMEDES left off, followed by this thesis' proposed improvement for the transport of FTV content over practical networks. This improved transport of FTV over networks is called OLFVmv (Optimized Live Free Viewpoint multiview video).

The DIOMEDES standards proposed a combined DVB/DTH (Direct Video Broadcast/Direct to Home) system combined with a P2P (Peer-to-Peer) network [DIOMEDES D2.3 2012] as further shown below:

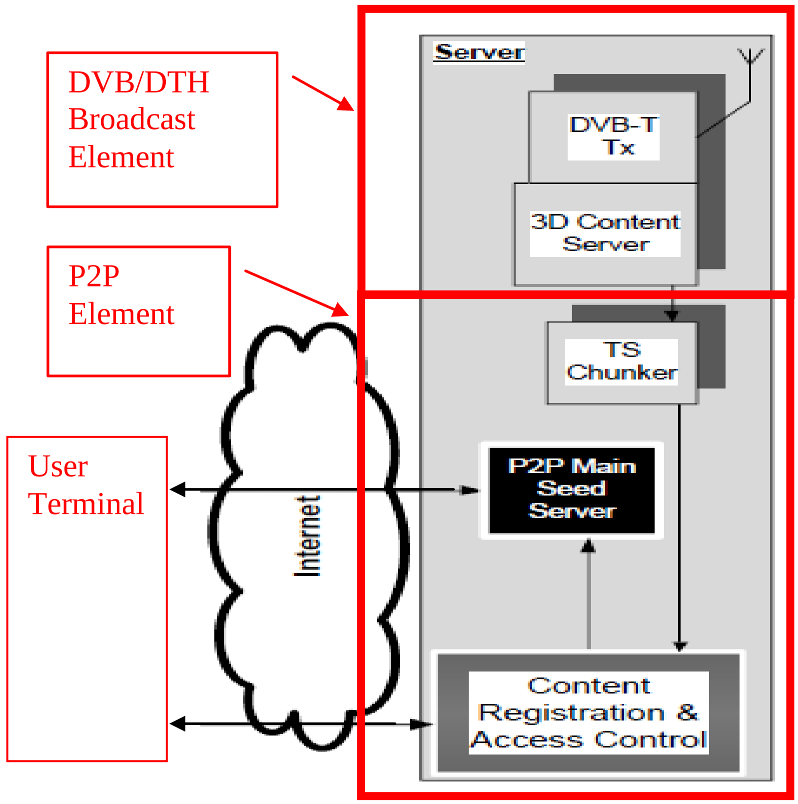

Figure 1: DVB + P2P FTV Architecture
(DIOMEDES std. D2.3 at pg. 8 [DIOMEDES D2.3 2012], annotated).

As shown in Figure 1, the DIOMEDES system provided a DVB-T (Direct Video Broadcast – Terrestrial) wireless stream and a "P2P Main Seed Server" network interface to the User Terminal.

The functionality of the P2P Main Seed Server in DIOMEDES was five-fold, as follows ([DIOMEDES D2.3 2012] at pgs. 12-13):

1) To provide a Bit-Torrent-Based protocol (discussed further below) that was used to transport content to a limited number of user terminals within a swarm.
2) To provide tracker software that enabled new peers to find existing peers.

3) To provide metadata that was used to initialize peers.

4) To provide authentication for peers.

5) To provide a PCR (Program Clock Reference) sanity check (e.g., to make sure the P2P content was reasonable synchronized and associated with the DVB transport stream).

Notably in DIOMEDES, the P2P Main Seed Server did not provide a feedback mechanism related to the desired content from the user terminals.

The DIOMEDES system focused not only on FTV but also on basic ("3D MVV" (3D Multiview Video) also called "2D plus delta" or "stereo" Multiview Video). Further, DIOMEDES employed a Scalable Video CODEC ("SVC" – *see* "Chapter 2: Background" of this thesis for a complete overview) approach to encoding the video, using the DVB channel transport to transmit the essential header and base layers, while using the DVB channel transport and/or P2P channel transport to deliver the non-essential enhancement layers [DIOMEDES D4.5 2011] as shown below:

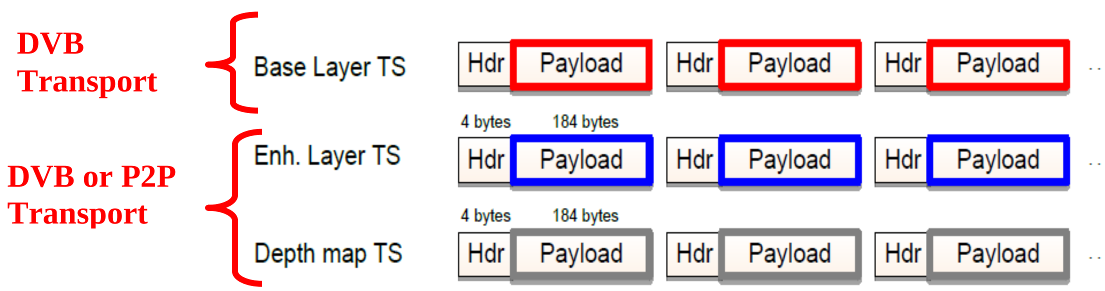

Figure 2: Splitting of content into multiple Transport Streams (TS)
(DIOMEDES std. D4.5 at pg. 12 [DIOMEDES D4.5 2011] (annotated)).

In DIOMEDES, the above process occurs N times for N separate video streams. For the P2P connection, a Bit-Torrent-Based protocol was used, with some important modifications including the splitting of chucks into GOP (Group of Pictures) boundaries, and the use of an "adaptation decision engine module (ADEM)" ([DIOMEDES D2.3 2012] at pgs. 32, 35) within the user's terminal. Because the

DIOMEDES ADEM is *within the user terminal, the ADEM functionality does not impact the overall selection and prioritization of video content distribution from the "Server" (see Figure 1).* The user terminal ADEM adapts the transport of three (3) camera views to the end user using the abovementioned DVB/DTH + P2P hybrid transport solution, as is further shown below:

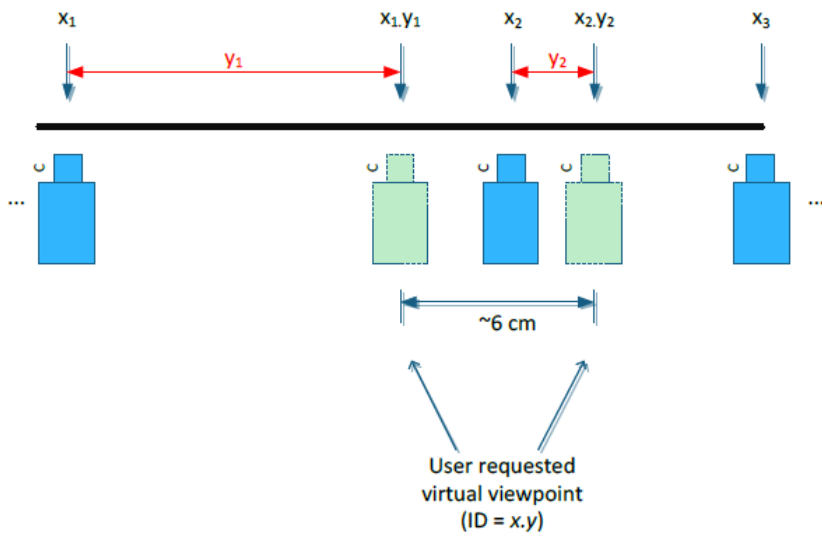

Figure 3: Rendering of virtual views from real views X1, X2, and X3
(DIOMEDES std. D3.6 at pg. 44 [DIOMEDES D3.6 2011]).

Using three (3) camera views, a DIOMEDES user was allowed to request and receive "user requested virtual viewpoints", "ID = x, y" from real camera views X1, X2, and X3. The ADEM module then prioritizes the GOP chucks over the P2P network (with no feedback to the DVB channel) as follows:

Chunk priorities P1-7 assigned to 3 camera group V1-3

View priority order (View-ID)	V1	V2	V3	V4	V5	V6	V7	V8
Base PID	P=1	P=3	P=4	P=8	P=10	P=12	P=14	P=16
Enhancement PID	P=5	P=6	P=7	P=9	P=11	P=13	P=15	P=17
Depth PID	P=2	P=3	P=4	P=8	P=10	P=12	P=14	P=16

Figure 4: Prioritization of GOP chunks over transport streams
(DIOMEDES std. D3.6 at pg. 45 [DIOMEDES D3.6 2011] (annotated)).

The literal (actual) camera views were called "core camera views". From these core camera views, the transport stream traffic was sent to the user's receiver to generate a virtualized view aligned to GOP boundaries.

1.2 Needed Improvements / Weaknesses with the Previous Solutions

While DIOMEDES made significant progress related to FTV system implementation, the DIOMEDES implementation was impractical for real-world systems; in the real-world, there is limited bandwidth for both the DVB broadcast channel transport as well as for the P2P network transport.

The excessive bandwidth requirements of FTV is compounded by the fact that DIOMEDES requires three core camera views over the broadcast DVB channels for each user to create a virtual view, where at best the DIOMEDES prioritization is based on the current view and the two surrounding, left and right, core camera views. The excessive bandwidth problems are further compounded on the P2P network which likewise seeks to transport three core camera views of the less important metadata depth map and/or enhancement layer information. Further as shown in Figure 4 above, the DIOMEDES system assigned priorities to all eight camera views, further making such a solution commercially impractical due to the amount of system throughput required.

1.3 Contribution of this Dissertation

In this thesis, an improved solution called OLFVmv (Optimized Live Free Viewpoint multiview video) is provided for the efficient and practical transport of live FTV. For a large population of viewers, it *is most cost effective to deliver content via a broadcast delivery transport such as DVB, as compared to P2P networks*. Taking advantage of this fact and building on what DIOMEDES has accomplished, OLFVmv provides the following improvements over previous solutions:

1) Intelligent algorithms combined with (a) a feedback mechanism to optimize the broadcast transport content efficiency, (b) requiring at most, two DVB channels, and

2) A roadmap to future work and improvements including the use of Network coding[1] for non-critical, video enhancement data to be sent via P2P networks.

The abovementioned improvements are accomplished by the further addition of two new system components as compared to the DIOMEDES system:

1) A new "P2P-Management Server" ("P2P-MS") to perform the abovementioned algorithms and provide feedback to the DVB broadcast system, and

2) A "DVB-Content Management Server" ("DVB-CMS") to receive input from the P2P-MS on the most prevalent (requested views) video content to broadcast, and based on that input, to select the most prevalent content to be broadcast over the two DVB broadcast channels.

Thus providing a solution for high-quality FTV content to be transported over existing DVB broadcast and P2P networks within practical real-world limits.

To expand, OLFVmv, utilizes a probabilistic model of the future viewed cameras based on monitoring a population of viewed cameras and establishing a prioritized trend of desired future views. Tracker software on the P2P-MS, which then applies an algorithm to predict a trend of cameras to be viewed. For example, turning to Figure 5, shown below, by comparing the viewing trend of users "USER1" through

[1] To expand, the type of network coding that will be adopted in this thesis is Hierarchical Network Coding (HNC) [NguyenNguyenCheung2010]

"USER N", a viewing trend is observed as transitioning from core camera views "V1" and "V2" at time state "T1" to "V2" and "V3" at time state "T4".

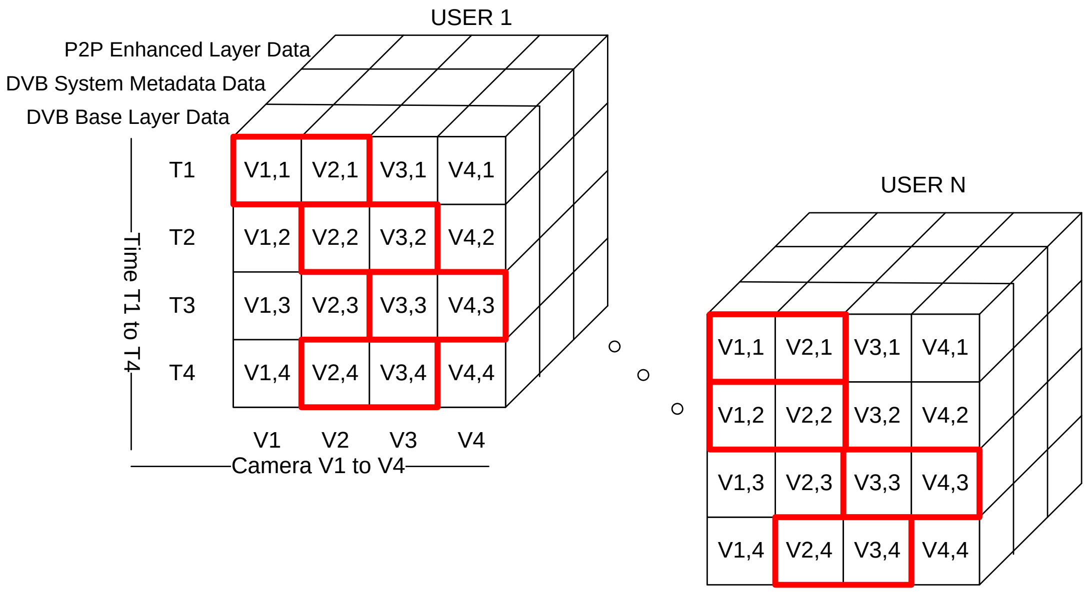

Figure 5: Proposed FTV content layering and viewer tracking

Further from Figure 5, it can be seen that the content for all core camera views are sub-divided into SVC layers including the system-wide metadata layer (including 3D/MVV information), base layer and enhanced layer. Because the proposed solution utilizes at most, two DVB[2] channels to broadcast content, the base and metadata layers and perhaps the enhanced layer can be sent via the DVB broadcast medium.

For discussion purposes:

> **Integer "Vx" is defined to be a specific primary core camera view for a user, called the *left* core camera view, and "Vx+1" is the adjacent *right* core camera view of "Vx".**
>
> **Non-Integer "Vn" is defined as a synthetic (simulated) desired view from the combination of the two left and right adjacent core camera views.**

[2] Via standard HDTV high definition TV methods.

As further shown below in Figure 6, a wide variety of intelligent algorithms can be developed to predict a viewing trend and then provide feedback to both the DVB broadcast and P2P transport mediums to layer and transport the FTV content into one of three transport layer modes:

1) Base layer and metadata layer content transport of the most desired Vx and Vx' views over the DVB broadcast medium.

2) Enhanced layer content transport of the most desired Vx and Vx' views over the DVB broadcast or P2P medium.

3) Base layer, metadata layer and enhanced layer transport over the P2P medium for non-predicted views as requested by specific users.

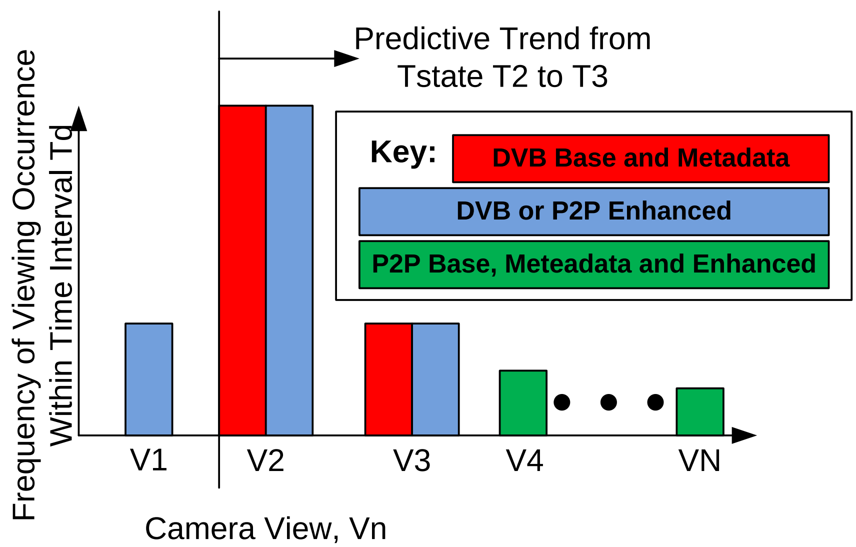

Figure 6: OLFVmv viewing trend modeling

Feedback is to be processed on the abovementioned P2P-MS and the trending desired channels will be provided to the DVB-CMS to select the most efficient views to provide over the DVB-T broadcast channels in real-time.

Additionally, in future work, it is envisioned that the P2P-MS will implement Hierarchical Network Coding [NguyenNguyenCheung2010] for the remaining desired views as requested by individual users to be transported over the P2P channel.

1.4 Simulation / Testbed Results Summary

Simulation and testbed results are provided in Chapter 4: Simulation / Testbed Results, followed by Appendix C – Matlab Simulation Code and Results. The simulation and testbed results provide a comparison of the priority based DIOMEDES content transport as compared to OLFVmv transport over a variety of simulated video content inputs and end viewer view requests. To expand the following results are provided:

1) **Baseline performance:** Baseline performance is based on the bandwidth utilization of the DIOMEDES like system with three channels transported over the DVB medium as DVB bandwidth allows, with all other content sent over the P2P network.

2) **Comparison performance:** By comparison, OLFVmv's performance results are provided using two full DVB HDTV channels for both base layer and metadata layer transport and the P2P network to transport the remaining data.

3) **Input variables include:** The testbed results are based on a variety of varying viewer viewing patterns for synthesized views (Vn) including the introduction of randomness between viewers and varying bandwidth conditions.

4) **Output variables include:** The bandwidth utilization and decodable received content from a simulated user is contrasted between the abovementioned DIOMEDES system simulation and the simulated results for OLFVmv.

Chapter 2: Background

A full overview of video encoding, compression technology, and MVV is separately presented by the author of this thesis. See reference [Kramer2016] of this thesis: "Richard A. Kramer, 'An Introduction to the Problem: Interactive Free Viewpoint Live Multiview Video Streaming Using Network Coding', Oregon State University, 2016." Subsequently, this Section is not intended to provide a full overview of video encoding and compression technology, but rather points to the specific attributes of MVV encoding that this thesis relies on. For a full primer on video encoding and compression technology, see reference [Kramer2016].

2.1 SVC (Scalable Video CODEC) Overview

MPEG-4 SVC (Scalable Video CODEC) allows for the dynamic video quality reception based on differing receiver input bandwidths across an entire system [Polycom2010][UittoVehkapera2013]. For example, SVC content includes both CGS (Course-Grain Quality Scalable Coding) and MGS (Medium-Grain Quality Scalable Coding) [Rimac-DjljeNemčićVranješ2008]. Thus, one device with limited bandwidth connectivity may only receive the CGS content, while yet another device with high bandwidth connectivity may receive both the CGS and MGS content.

The SVC layering of MVV/FTV information is a fundamental aspect that enables the MVV/FTV information to be subdivided into independent transport streams over the DVB and P2P transport mediums.

Because a clear goal of OLFVmv is to reduce bandwidth of the various content streams, unlike DIOMEDES that seemingly focuses exclusively on DVB-T, the goal of OLFVmv is to support a wide variety of DVB transports including cable/satellite systems [ETSI EN300 429 V1.2.1:1998][ETSI EN200 421 V1.1.2:1997][ETSI EN302 307:2009], handheld device based systems that offer far less bandwidth [ETSI EN302 304 V1.1.1:2004], DVB over IP networks as a whole [ETSI TS102 034 V1.4.1:2009], and the second generation DVB-T (Digital Video Broadcast - Terrestrial) systems, called DVB-T2 [ETSI EN 302 755 V1.2.1:2010]. Therefore in conjunction with OLFVmv, this thesis' referral to "DVB" and in not solely limited to DVB-T. This thesis builds on the known transport of MVV over IPTV (Internet Protocol Television) [SchielNarashimhan2011] as well as the transport of SVC based video content via handheld mobile IPTV devices [KimLee2011].

2.2 MVV (Multiview Video) Standardization

The first MVV standard was approved in July 2008 ([Ohm2009] at pg. 9). The MVV Standard was formally integrated into the Fifth Edition of the overall MPEG-4 Standard (H.264/MPEG-4 Std. ISO/IEC 14496-10) as Annex H [ISO/IEC 14496-10:2008]. The 2008 MPEG-4 Standard [ISO/IEC 14496-10:2008] was further revised in 2009 to add "Multiview High Profile" [ISO/IEC 14496-10:2009] which is the basis of FTV utilized by OLFVmv.

2.3 MVV (Multiview Video) Standardization Overview

Within the abovementioned MVV standards, MVC (Multiview Video Coding) is used to accomplish the layering and subdivision of the MVV information that further enables FTV transport of OLFVmv via the DVB and P2P transport mediums. From the MVV standards noted above, the following MVC *profiles* were standardized as follows:

1) <u>Stereo High Profile</u>. Stereo High Profile in the H.264/MPEG-4 standard, also known as "3D" and/or "Video-plus-Depth" and/or "2D plus Delta" entails the following:

- Primarily used for 3D movies including Blue-Ray.
- Various methods are employed to display 3D movies (3-D glasses at theaters, holographic displays, etc.).

2) Multiview High Profile. Multiview High Profile is the profile that is the subject of the thesis. The Multiview High Profile supports an arbitrary number of camera views, also known as "Free-viewpoint Video" or "FTV" (Free-viewpoint TV)

- FTV is used for example, to obtain differing views of a field in a sports competition, such as soccer.

Overall a summary of H.264/MPEG-4 for MVC is as follows:

Important H.264/MPEG4 / AVC Revisions related to MVC:

ISO/IEC 14496-10:2009: (March 16, 2009) Major addition to H.264/AVC containing the amendment for Multiview Video Coding (MVC) extension, including the **Multiview High profile**. [ISO/IEC 14496-10:2009].

ISO/IEC 14496-10:2010 (March 9, 2010) Amendment containing the definition of the **Multiview Stereo High profile** for two-view video coding with support of interlaced coding tools and specifying an additional SEI message (the frame packing arrangement SEI message). [ISO/IEC 14496-10:2010].

ISO/IEC 14496-10:2014 (August 27, 2014): Amendment to specify the coding of depth map data for 3D stereoscopic video, including a **Multiview Depth High profile**. [ISO/IEC 14496-10:2014].

The Stereo High Profile and Multiview High Profile are described in more detail as follows:

2.4 Stereo High Profile

As shown in Figure 7 below, for Stereo High Profile MVC, two cameras are placed with FOVs (Field Of Views) that are parallel to one another. Based on the parallel camera FOVs, a "depth map" is generated by comparing the depth perception afforded by differing stereo camera views of the same object(s). In short, the depth of objects in a scene can be calculated mathematically using the stereo camera views.

Referring to Figure 7 below, a synthesized (virtual) 3-D view can then be generated using a baseline video camera view combined with depth map information to generate a synthetic view.

Highlight attributes of Stereo High Profile MVC include:

- At minimum: one video, one depth map
- **Technologies required:**
  - Depth estimation.
  - Depth encoding.
  - View synthesis using encoded image and depth map.

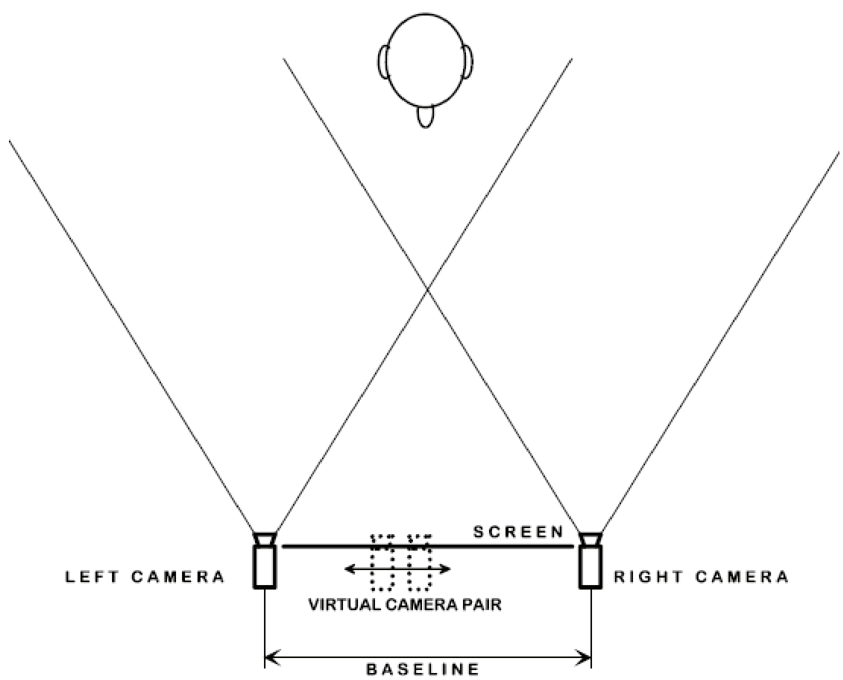

Figure 7: Stereo High Profile / 3D Video ([Ohm2009] at pg. 6)

An example of an encoding and transmission system for Stereo High Profile MVC is shown in Figure 8 below. As shown in Figure 8, the "left view" is used to generated the primary video information and the "right view" is used solely to generate "depth image estimation" information. Thus for the right view, only P-Frame and B-Frame prediction information is transmitted. The resulting video transmission bandwidth savings is the difference in bandwidth between transmitting the complete right view in a video compressed encoded form, versus simply transmitting the depth image estimation information generated by depth image encoder (shown below).

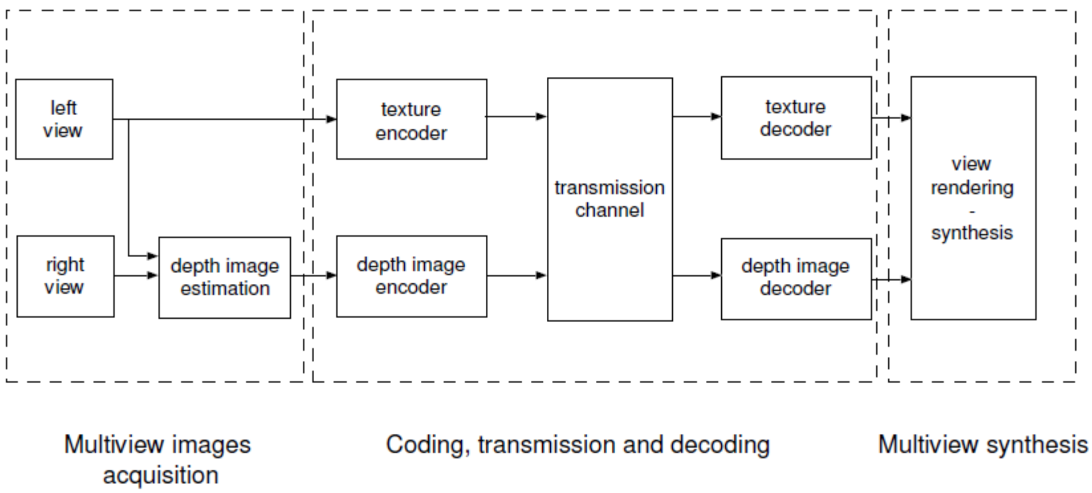

Figure 8: Example of Stereo High Profile MVC image transmission system [Morvan_deWithFarin2006]

Advances in overall video compression technology have been introduced per the ITU-T Recommendation H.265 and ISO/IEC 23008-2 (MPEG-H) which employs HEVC (High Efficiency Video Coding). HEVC has been extended for stereoscopic MVC offering up to 50% in bandwidth savings overall as compared to H.264 AVC.. Additional proposals have been made regarding further bandwidth savings based on motion estimation encoding that extend beyond HEVC [MüllerSchwarz2013].

2.5 Multiview High Profile

For Multiview High Profile MVC, multiple cameras are used in a parallel array (*see* Figure 9, below at left). What the user sees on a display is either part of an actual image, or a synthetic image that is created by a combination of other images. Figure 9 below, shows such a camera array and the transmission bandwidth savings of sending each camera view separately (e.g., simulcast) versus sending the views via Multiview High Profile MVC encoding (e.g. Multiview).

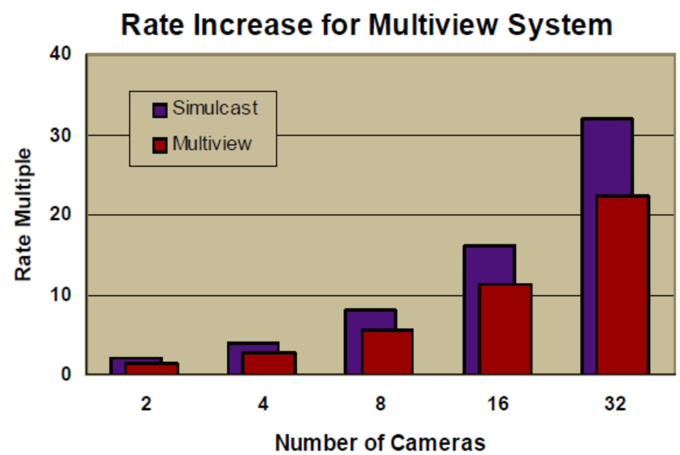

Figure 9: MVC standard – limitation/issues ([Ohm2009] at pg. 14)

As shown in Figure 9, the "Multiview" transmission bandwidth savings is a result of the Multiview High Profile MVC encoding techniques which are further explained in this Section. Overall, the video camera image data between adjacent parallel camera views in an FTV system contains large amounts of inter-view statistical dependencies, therefore those dependencies can be exploited using standard MPEG-4 inter-frame encoding techniques.

To expand, as shown in Figure 10, below, the only *full* spatially compressed image (e.g., an "I-Frame" (Intra-Frame)) is for "Cam 1". Thus only the video encoder for Cam 1 requires the transmission bandwidth for transmission of a full spatially I-Frame compressed image. For the remaining camera views "Cam 2" through "Cam 5", the video encoders for each camera takes the full camera image (of Cam 2 through

Cam 5), and encodes either a "B-Frame" (Bi-Directional-Frame) or "P-Frame" (Prediction Frame) by exploiting the statistical similarities of information in the adjacent camera images.

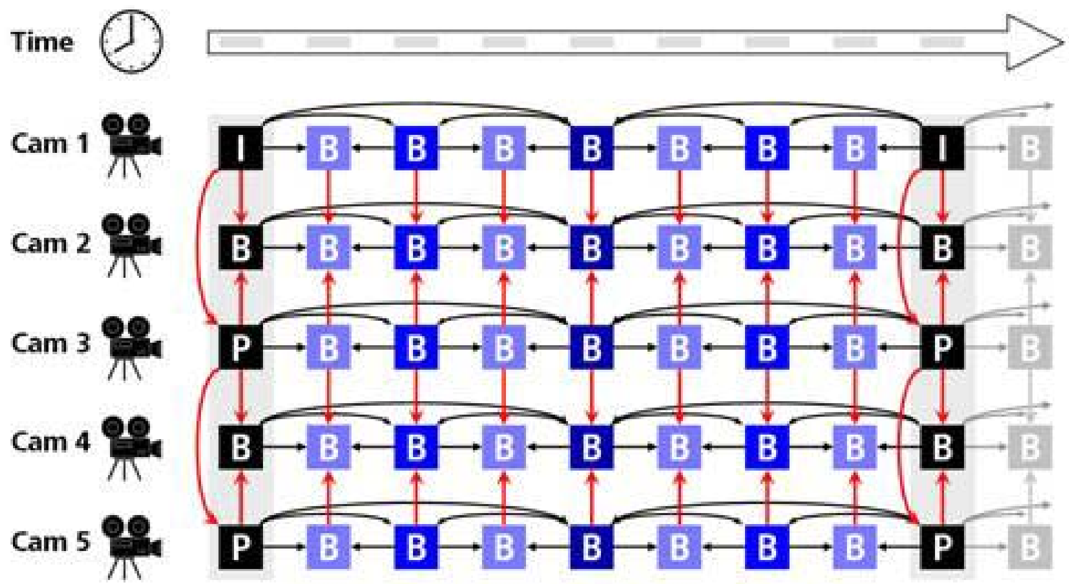

Figure 10: Temporal P and B-Frame prediction structure for MVC [Smolic2008, OhmSullivan2005]

Likewise, as also shown in Figure 10, each camera view employs standard MPEG-4 video compression, and thus after the initial frame, each camera video encoder generates a stream of frames over time such as I-B-B-B-B-B-B-B-I, as shown for Cam 1 above. The subsequent B-Frames or P-Frames afford additional transmission bandwidth savings over the initial I-Frame.

Highlight attributes of Multiview High Profile MVC include:

- At minimum: two video views.
- **Technologies required:**
  - Intra (I-Frame) and Inter (B-Frame and P-Frame) Frame encoding between images.
  - View synthesis using resulting encoded image.

2.6 Video Compression Data Layering

The inter and intra frame video compression information as described above is encoded into layers. Video information that is unique to an image is encoded as macroblocks at the Video Coding Layer (VCL). The VCL layer information is then input to the higher Network Abstraction Layer (NAL) as shown in Figure 11, below, and transported over the transport medium.

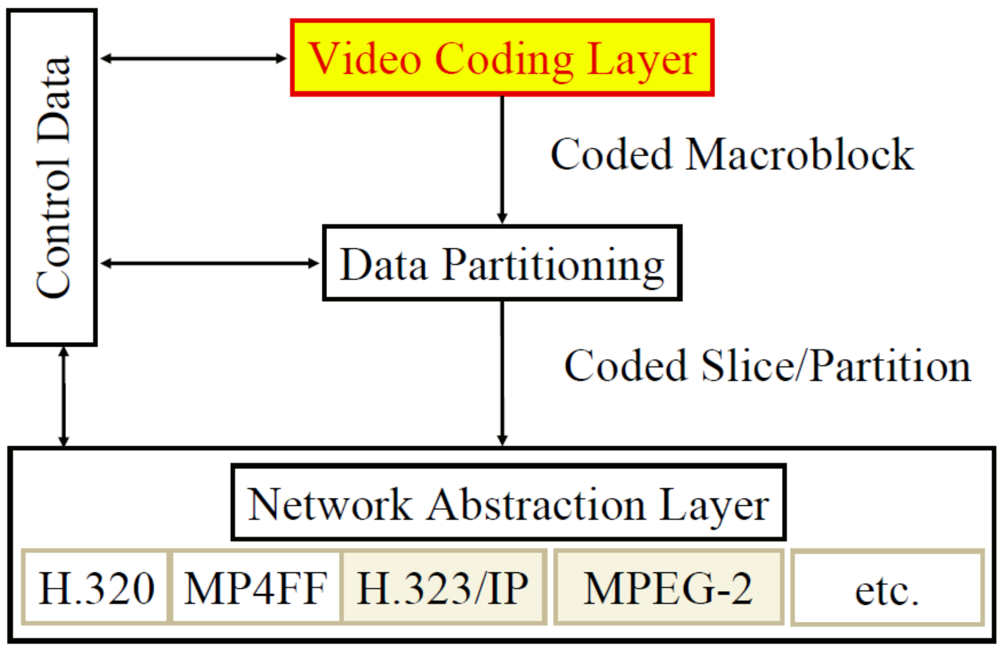

Figure 11 – Video compression data layering [WiegandSullivan2003]

The key attributes about the data layers that are shown above include:

- NAL (Network Abstraction Layer) messages are call "units".
- There are multiple "types" of NAL units that convey both VCL and non-VCL information.
- Each NAL unit type is called an NAL Unit Type ("NUT").

From the above, the video information is subdivided into "VCL" and "non-VCL" information. While VCL information is specific to a macroblock of a specific camera video image, non-VCL information can be common information shared across some (or all) of the FTV camera video views/images (such as video compression coefficients ([Kramer2016] at pgs. 11-13).

Specifically related to MVV encoding, as further shown in Figure 12, the MVV data is layered into "Base View" units and "Non-Base View" NAL units. As shown below, for the Base View, a coded video sequence always starts with an Instantaneous Decoding Refresh (IDR) access unit, which signals that the IDR access unit and all access units that follow it in the bitstream can be decoded without decoding any of the pictures that preceded it (*see* Figure 12, light gray shaded boxes which is information resulting from VCL information). Examples of non-VCL information (see Figure 12, dark gray shaded boxes) include Sequence Parameter Set (SPS) information as defined by H.264/MPEG-4 AVC [WiegandSullivan2003]. This information includes shared information across all camera views, including supervisory (metadata layer) information.

Three important pieces of information are carried in the SPS extension:

- View identification.
- View dependency information.
- Level index for operation points.

Importantly, as shown below, the information is subdivided into "Base View" and "Non-Base View" NAL units. Base View units can be decoded as standard AVC (or SVC) MPEG-4 (non MVV) video decoders, thus affording broadcasting.

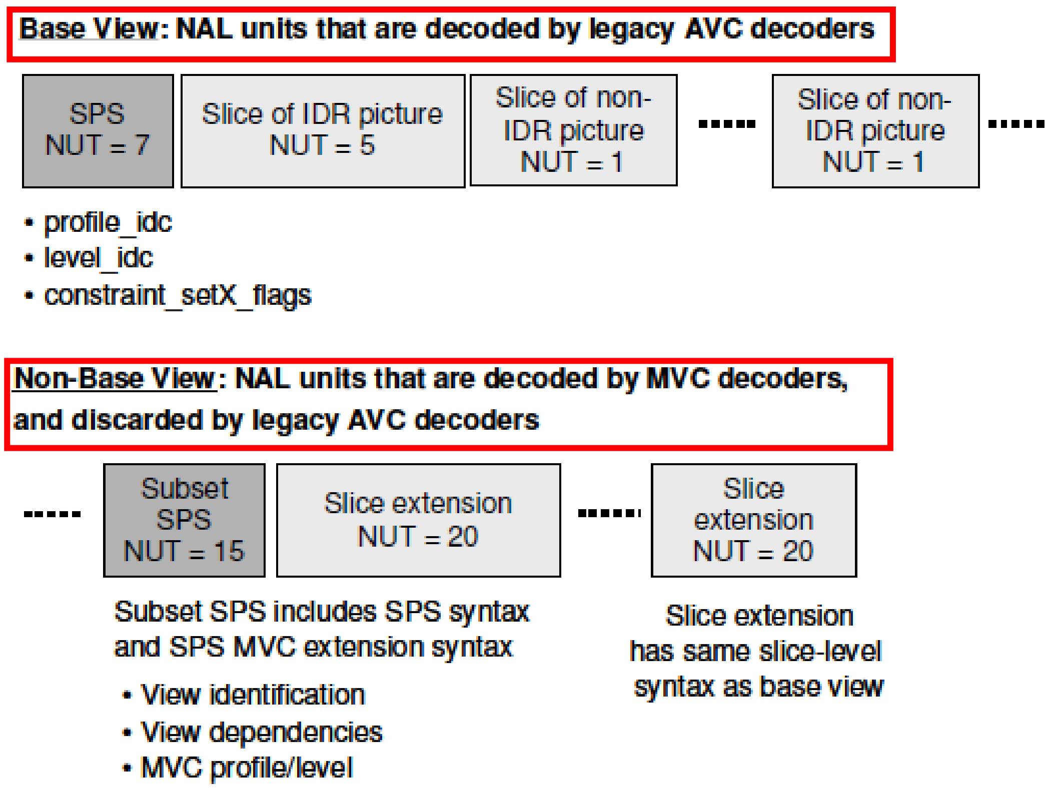

Figure 12: Structure of an MVC bitstream including NAL units [VetroWiegandSullivan2011]

From this, MVC exploits significant sharing of common information between views. As further shown in Figure 13, below:

- Common (non-VCL) information for all views can be sent via a separate communications apart from the VCL data:
  - SEI (Supplemental Enhancement Information)
  - Parameter Sets

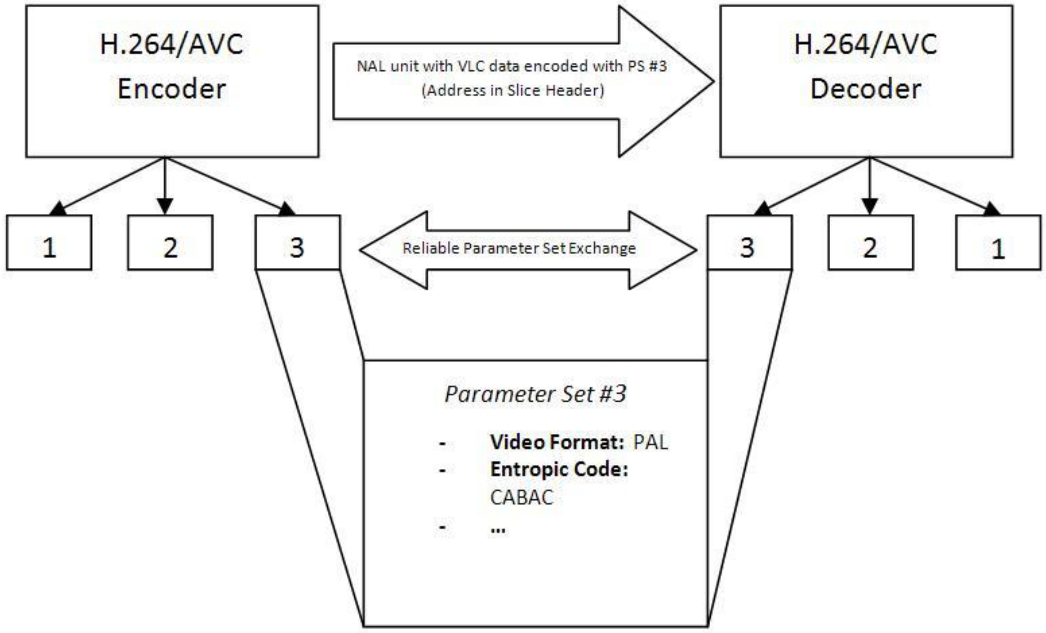

Figure 13: Transport of VCL and Non-VCL video information

2.7 Background Summary

From the above, significant opportunities exist to optimize MVC encoding and improve transport technologies for FTV. For example:

- The layering of MVC encoded information, combined with Network Coding provides a fertile area of research to optimize network efficiency over both DVB (broadcast) channels and P2P (network) channels.
- Specifically, the opportunity exists to maximize bandwidth efficiency for multiple users within latency, bandwidth limitations and bandwidth error constraints.

Chapter 3: Optimized Live Free Viewpoint multiview video (OLFVmv)

3.1 Improved FTV MVV System Architecture

This Chapter discloses an improved FTV MVV System architecture. Figure 14, shown below, shows a modified DIOMEDES architecture that includes:

- A new DVB – Content Management Server (DVB-CMS). The new DVB-CMS consists of both server hardware and specialized software that runs on the DVB-CMS server hardware. *The new DVB-CMS is described in Section 3.2 of this Chapter.*

- A new P2P-Management Server (P2P-MS). The new P2P-MS consists of both server hardware and specialized software that runs on the P2P-MS server hardware. *The new P2P-MS is described in Section 3.3 of this Chapter*.

In this thesis, a FTV MVV system that contains four and eight core camera views is considered, however, any number of core camera views can be used. The choice of four and eight core camera views is simply chosen for simplicity.

(continued on next page).

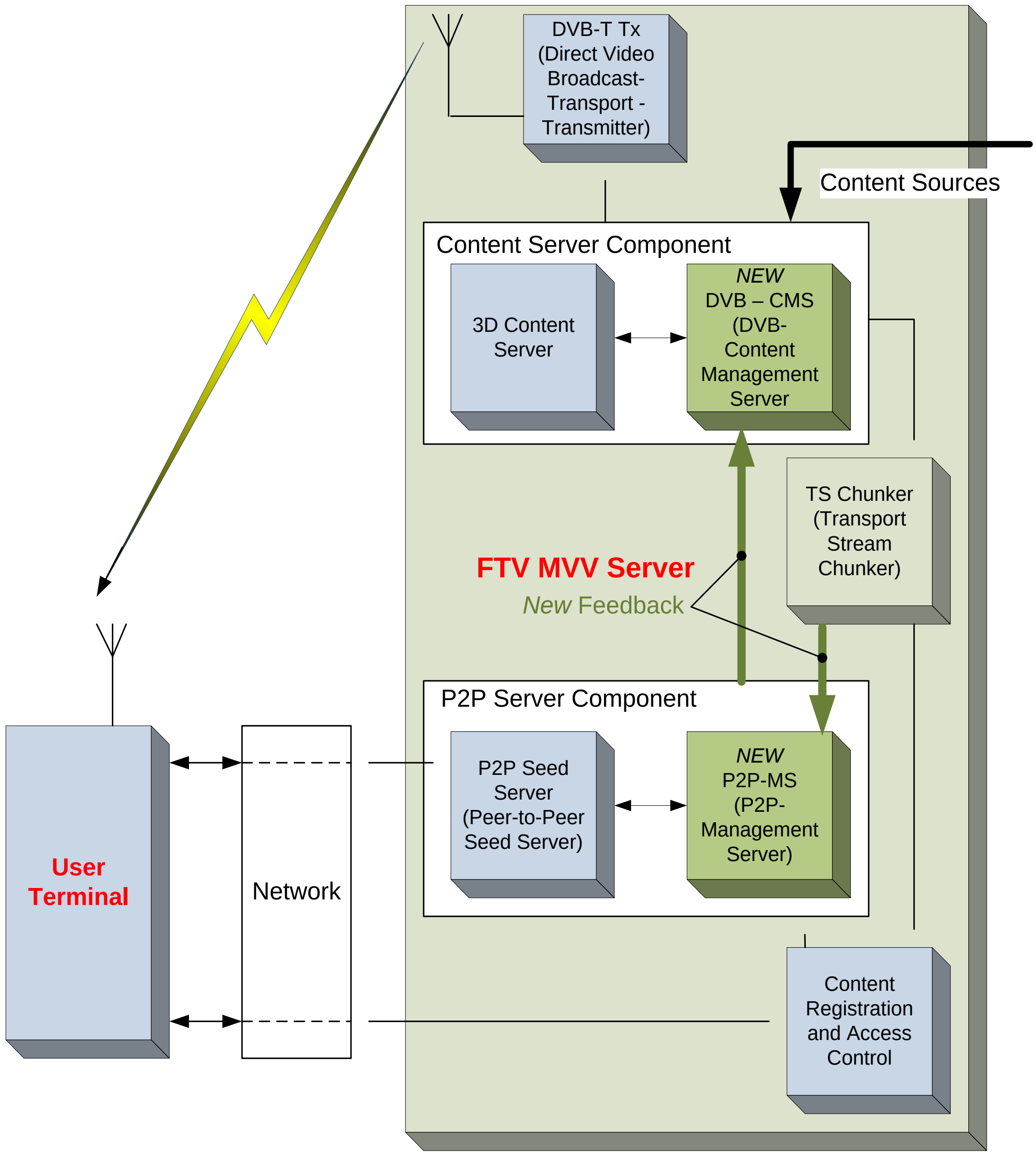

Figure 14: Optimized FTV MVV System with FTV MVV Server

3.2 DVB–Content Management Server (DVB-CMS)

Of the improvements offered by OLFVmv over previous FTV proposed systems is the addition of the DVB–Content Management Server (DVB-CMS henceforth). The DVB-CMS is either a separate server or is contained within the overall FTV MVV Server as shown above in Figure 14. The DVB-CMS software that runs on the DVB-

CMS selects the two most prevalent core camera views, based on the newly added OLFVmv system feedback received from the P2P Server Component (to the Content Server Component) as further shown in Figure 14 (*see* "Feedback"). Based on the feedback from the P2P Server Component, the DVB-CMS streams two core camera views over the DVB channel using standard DVB transmission means using MPEG-4 SVC, and further capitalizes on the adjacent view redundancies to use P and B Frames as shown in Figure 10. As shown in Figure 15, below, both the "DVB Base Layer Data" and "DVB System Metadata Data" are transported via the DVB medium.

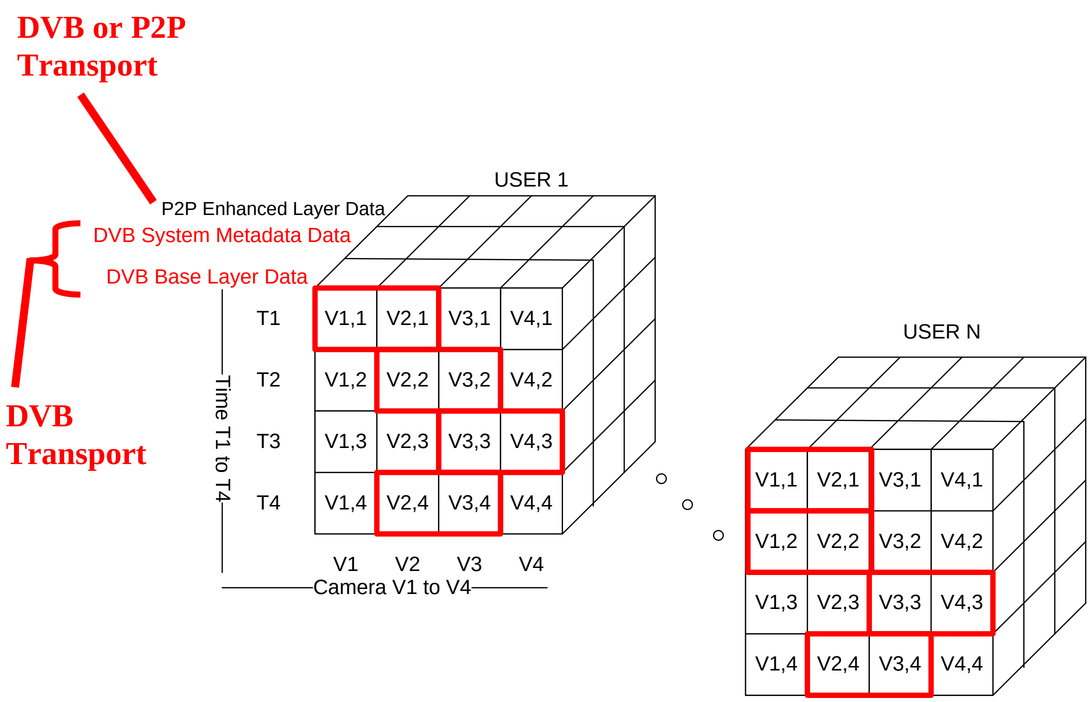

Figure 15: DVB FTV content delivery

3.3 P2P-Management Server (P2P-MS)

An additional improvement that OLFVmv offers over previous systems is the addition of the P2P Management Server (P2P-MS henceforth) and the additional feedback and processing provided by the P2P-MS related to intelligent video content prioritization and prediction.

The P2P-Management Server is either a separate server or is contained within the overall FTV MVV Server as shown above in Figure 14. The P2P-MS software that runs on the P2P-MS is the primary component of the system described herein. In short the P2P-MS provides the following functions:

1) Collects user viewing trend data of what primary core video camera views are being watched from users' terminals and from these inputs (e.g. viewing trends), records aggregate viewing trends for the overall system (*see* Section 3.4.1 P2P-MS Viewing Trend Mapping).
2) From the recorded system-wide user viewing trends, executes an algorithm that selects the most prevalent (and thus most efficient to transport) DVB and P2P network video content transmission needs (*see* Section 3.4.2 P2P-MS to DVB_CMS Video Content Selection Algorithm).
3) Selects DVB video content to be transmitted over the DVB channel and provides this feedback directly from the P2P-MS (P2P Server Component) to the DVB-CMS (DVB Content Server Component) (*see* Section 3.4.3 DVB Video Content Selection).
4) Based on what the DVB channel is able to transmit, the TS (Transport Stream) Chunker then informs the P2P-MS (P2P Server Component) of the residual content that was not able to be transmitted via the DVB channels (see Section 3.4.4 P2P-MS Network Video Content Selection Feedback).
5) Selects Network video content to be transmitted over the P2P network channel (see Section 3.4.5 P2P-MS Network Video Content Selection Prediction).
6) Applies Hierarchical Network Coding and Network Management [NguyenNguyenCheung2010] to the selected network video content that is transmitted over the P2P network channel (*see* Section 3.4.6 P2P-MS Hierarchical Network Coding and Network Management of the P2P Selected Network Video Content).

3.4 DVB–Content Management Server (DVB-CMS) and P2P-Management Server (P2P-MS) Video Content Processing

One of the key aspects of both the DVB-CMS and the P2P-MS is the ability of these servers to intelligently process and select the most important video to transport via the respective DVB and P2P channels. This section describes the methods employed by both the DVB-CMS and P2P-MS in detail.

3.4.1 P2P-MS Viewing Trend Mapping

A critical aspect of the overall FTV MVV System is the ability of the P2P-MS to collect user viewing trends in real-time and report those viewing trends back to the DVB-CMS. For each user terminal, the real-time Vn view's position is registered. Based on the position of Vn, for user n ∊ N (where N is the total viewer population), the two adjacent core camera views, Vn_Left and Vn_Right, are registered into a matrix for all users in the system that tracks the view positions.

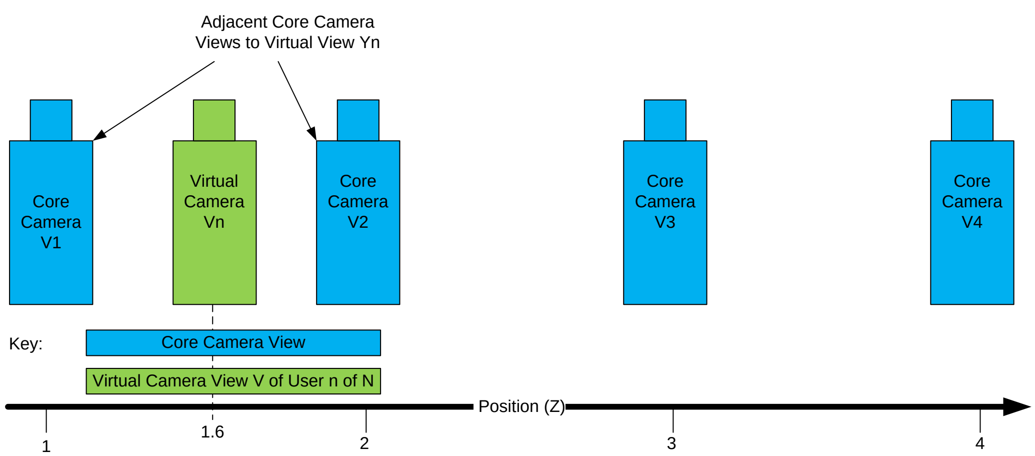

Figure 16: Core camera registration based on virtual camera view Vn.

For the example for each user n:

Vn_Left = The left adjacent camera view to user n's current viewing position Vn.

Vn_Right = The right adjacent camera view to user n's current viewing position Vn.

Thus, for the above shown system (*see* Figure 16, above) with Core Cameras = 8 cameras and Users = 100 users, the example algorithm below reads each user's current viewing position (e.g., what virtual camera position they are looking at, "Vn") and from that, assigns the Left and Right cameras. The below algorithm[3] uses a random generator for each user to simulate a diversity of viewing positions among users:

See Appendix C for a complete list and explanation of variables.

```
%% Camera view registration algorithm

function register_cameras_views

global Num_Cameras
global Vn_Viewer_View
global Viewer_Time_Osc_Position
global Viewer_Random_Position_Offset

% Virtual desired view = mean viewing position, plus/minus random offset
% This operation takes the offset matrix [Num_Viewers x 1] and adds to the
% View Position matrix [1 x Time Ticks] and equals a view position for each
% viewer = [Num_Viewers x Time Ticks] matrix

Vn_Viewer_View.Vn ...
   = min(max((Viewer_Time_Osc_Position +
Viewer_Random_Position_Offset),1),Num_Cameras);

% Calculate left most camera by translating Vn in to an integer with the
% minimum camera being Camera = 1 and the right most camera being one from
% the, Camera = Num_Cameras - 1
Vn_Viewer_View.Vn_Left = floor(min(max(Vn_Viewer_View.Vn,1),Num_Cameras - 1));

% Calculate right most camera by taking Vn_Viewer_View.Vn_Left and adding
% one.  The case should never exist where the Right most camera exceeds
% Num_Cameras, but if it does, limit it to Num_Cameras
Vn_Viewer_View.Vn_Right = int8(min((Vn_Viewer_View.Vn_Left + 1),Num_Cameras));

fprintf('execution complete: register_cameras_views \n')

end
```

Figure 17: Camera view registration algorithm

[3] The software code presented in this thesis including in Appendix C is in Matlab.

It is notable that the above software assigns the left and right core camera views for each and every viewer, where Vn_Viewer_View is an array of size Num_Viewers (*see* Appendix C). The current viewer's camera view Vn provides FTV image synthization based on the adjacent Vn_Left[n] and Vn_Right[n] core camera views. It is also notable that if the user's current view is the rightmost core camera, then Vn_Left[N] and Vn_Right[N] is essentially decremented by one as not to exceed the core camera views which still allows a FTV view from a left viewing angle.

From the above example, the next step is for the P2P-MS to generate a histogram for all viewers whereas in this example, the number of users is N=10, as follows for a four camera system:

Table 1: Example of camera position data for N users.

User (n)	Vn Current View	Vn_Left[n] Left Core Camera	Vn_Right[n] Right Core Camera
1	1.6	1	2
2	1.2	1	2
3	1.5	1	2
4	2.1	2	3
5	1.7	1	2
6	2.0	2	3
7	1.7	1	2
8	1.3	1	2
9	4.0	3	4
10	2.0	2	3

(continued on next page).

From this, the following histogram is revealed for the left and right core camera views:

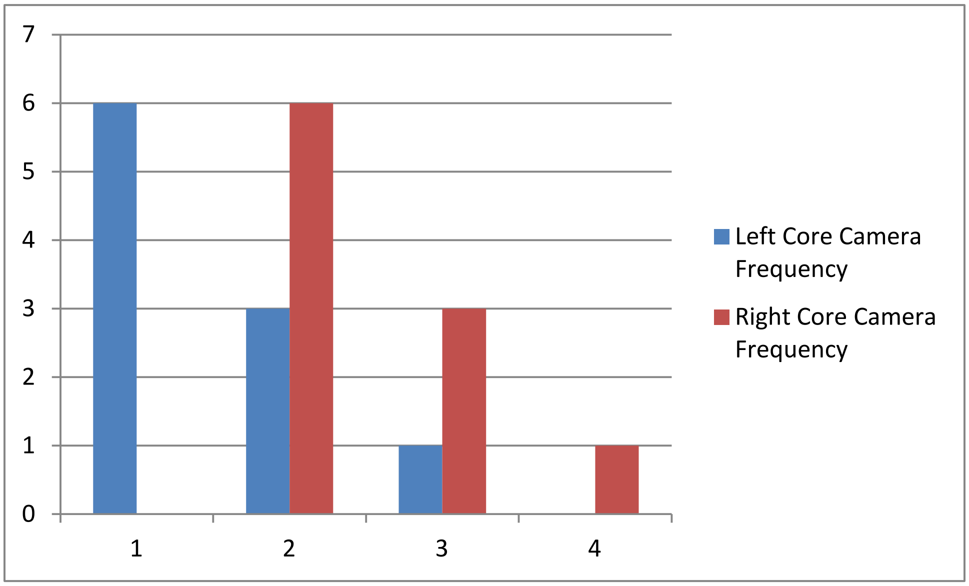

Figure 18: Core camera view histogram.

3.4.2 P2P-MS to DVB_CMS Video Content Selection Algorithm

From the above, the viewer statistics is then processed in an algorithm to find the most popular adjacent left and right core camera views. This algorithm may take many forms and range from simply finding the most popular views based on the above shown histogram (*see* Figure 18, above), to more elaborate prediction algorithms.

As an example of an algorithm that may be used, the below algorithm finds the two most prevalently viewed (left and right) core camera views as derived from Table 1 and Figure 18. These two most prevalent core camera views are then assigned with the highest priorities such that the related content is transmitted over the DVB medium transport (see Section 3.4.5 P2P-MS Network Video Content Selection Prediction, below).

By way of example as shown below, the simulation software code then builds the histogram for all viewers. Using this histogram, the most important (e.g., most demanded) video content can then be identified.

Continued from Figure 17, above[4].

```
%% Algorithm to find left and right core camera views to transmit over the DVB medium based
on building a histogram

function build_viewing_histogram

global Vn_Viewer_View
global Channel_Histogram
global Num_Sim_Run_Time_Ticks
global Num_Cameras

% define bins 0.5-1.5, 1.5-2.5, and so on to capture the center of each bin
% at 1, 2, 3, 4, ... Num_Cameras, camera views into Num_Cameras discrete bins
bin_edges = 0.5:1:Num_Cameras+0.5;

% for each time tick, do a histogram of camera number (Vn) viewed, by the
% total population of all viewers

for Histogram_Time_Index = 1:Num_Sim_Run_Time_Ticks
   Channel_Histogram(:,Histogram_Time_Index)...
      = histcounts(Vn_Viewer_View.Vn_Left(:,Histogram_Time_Index),bin_edges);
end % end - for

fprintf('execution complete: build_viewing_histogram \n')

end % end - build_viewing_histogram
```

Figure 19: Algorithm to build a viewing histogram

3.4.3 DVB Video Content Selection

As shown in the above Camera View Histogram for all users (*see* Table 1 and Figure 18), the most prevalent two core camera views that occur in the histogram are communicated in real-time from the P2P-MS to the DVB-CMS as shown in Figure 14, above. This information is immediately used by the Content Server Component.

[4] See Appendix C for a complete list and explanation of variables.

As a result, the two most prevalent core camera views within the histogram are directly broadcast to all users via the DVB medium for all layers of the HDTV signal.

Importantly, because the left and right core camera views are adjacent to one another, only P-Frames and B-Frames are needed for the right core camera view as is further shown in Figure 20, below.

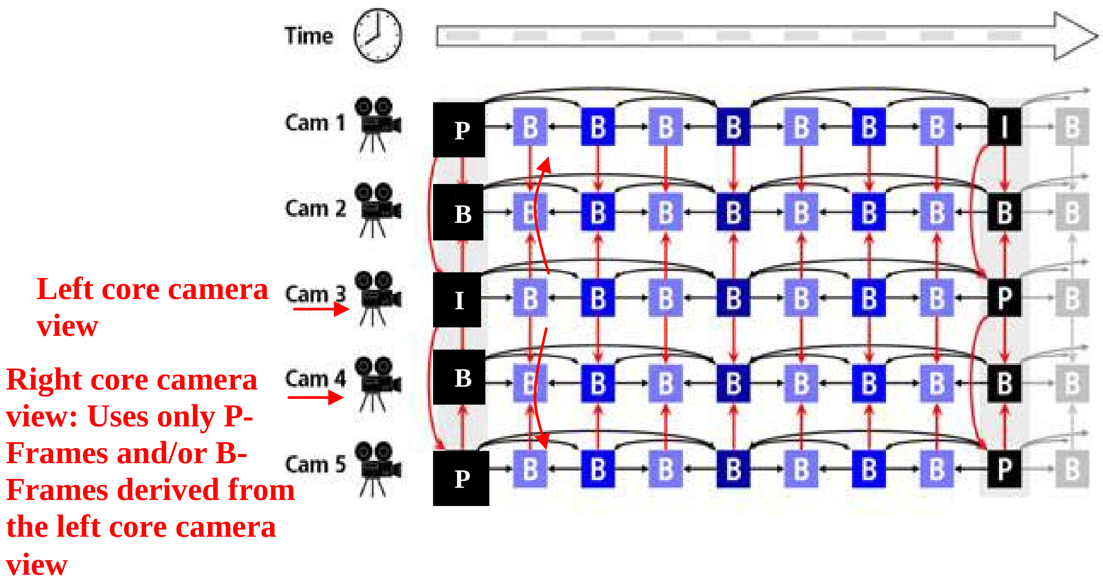

Figure 20: Temporal P and B-Frame view prediction structure for MVC

In the above Figure 20, the assumed most prevalent left core camera view is shown to be “Cam 3” and thus the right core camera view is “Cam 4”. Accordingly, the encoding of “Cam 4” is done using only P-Frames and/or B-Frames that are derived from the left core camera view, “Cam 3”. As the most prevalent core camera view moves, say from “Cam 3” to “Cam 4”, then a leading I-Frame will be generated for “Cam 5” and the right core camera view “Cam 5” will be accordingly encoded using P-Frames and/or B-Frames.

By using this process, a maximum of two HDTV channels are used by the DVB transport medium. Further, all users are provided the two most prevalently viewed core camera views using the most efficient transport medium (e.g., the DVB medium). Further, because only P-Frames and B-Frames are used for the rightmost core camera view, the DVB channel for the second view has additional capacity to transport the lower priority metadata layers and enhancement layers for the two most prevalently viewed core camera views.

3.4.4 P2P-MS Network Video Content Selection Feedback

While the transport of the DVB channels provide the full HDTV scalable video content for preferably all layers and at a minimum for the base and metadata layers (*see* Figure 15 above), the P2P channel is used for the less popular core camera layers and views. Further if the DVB channel does not accommodate the HDTV bandwidth requirements, the P2P channel can also transport the left and right core camera view enhancement layers that correspond to the DVB views.

(continued on next page).

All possible P2P channel content to choose from can be shown by the following expression:

Let $\mathbf{V_{ALL}} = \sum_{m=1}^{M}(V_{Meta}[m] + V_{Base}[m] + V_{Enhanced}[m])$

Where:

$\mathbf{V_{ALL}}$ = All possible video content (all camera views, and all layers),

And M represents the maximum core camera view.

Further, let: m_LEFT be the most prevalent left core camera view number, and let m_RIGHT be the most prevalent right core camera view number.

Then the video transported by the DVB transport medium is expressed as:

$$V_{DVB} = V_{Meta}[m\_LEFT] + V_{Base}[m\_LEFT] + V_{Enhanced}[m\_LEFT]) + V_{Meta}[m\_RIGHT] + V_{Base}[m\_RIGHT] + V_{Enhanced}[m\_RIGHT])$$

And the all possible video content to choose from for the P2P channel can be expressed as:

$V_{P2P} = \mathbf{V_{ALL}} - V_{DVB}$

Figure 21: Expression of all possible P2P channel video content

Accordingly, such information (e.g., V_{P2P}) is communicated from the TS Chunker to the P2P-MS, thus informing the P2P-MS of the total population of video content to choose from.

This is further shown in the following example, assuming an eight core camera system, with the most prevalent left core camera view being V3 and thus the right adjacent core camera view being V4:

Key:

V_{DVB} = BLUE	
V_{P2P} = GREEN	

Core Camera View →	V1	V2	V3	V4	V5	V6	V7	V8
Base Layer	P2P	P2P	DVB	DVB	P2P	P2P	P2P	P2P
Metadata Layer	P2P	P2P	DVB	DVB	P2P	P2P	P2P	P2P
Enhanced Layer	P2P	P2P	DVB	DVB	P2P	P2P	P2P	P2P

Figure 22: Example of video content distribution between the DVB and P2P channels

From Figure 22 above, the P2P-MS then uses the feedback of what the DVB medium was able to transmit, to then predict the most valuable information to send over the P2P network medium. The methods to do this are an improvement over DIOMEDES as is further explained in Sections 3.4.5 and 3.4.6 below.

3.4.5 P2P-MS Network Video Content Selection Prediction

Via the P2P-MS, video content to be transported over the P2P channel is prioritized. This prioritization entails two parts: First, a baseline priority must be established for the system based on the abovementioned histogram analysis. Second, the prioritization of the video content is then altered over time based on viewer trends to "look" left or right of the current histogram maximum left and right core camera views. This section further describes the prioritization process accordingly.

3.4.5.1 Baseline Video Content Prioritization

To fully appreciate the importance of prediction, the priority scheme of DIOMEDES prioritizes video content GOP chucks, from left to right, starting with camera core view V1 as shown:

View priority order (View-ID)	V1	V2	V3	V4	V5	V6	V7	V8
Base PID	P=1	P=3	P=4	P=8	P=10	P=12	P=14	P=16
Enhancement PID	P=5	P=6	P=7	P=9	P=11	P=13	P=15	P=17
Depth PID	P=2	P=3	P=4	P=8	P=10	P=12	P=14	P=16

Figure 23: Prioritization of GOP chunks over transport streams
(DIOMEDES std. D3.6 at pg. 45 [DIOMEDES D3.6 2011]).

The improvement of OLFVmv provides prediction based on viewing trends taking advantage of feedback from the viewers. While this feedback must be in real-time, there is only a small amount of data that is relayed back from the user terminals identifying what core camera views are most prevalent. From my direct experience in the industry including my development of PTZ (Pan Tilt Zoom) cameras controlled by users: a response time of 100 mS (and as much as a few hundred milliseconds) between when the user desires to move the camera view, as compared to the response time of when the actual image view moves, is more than acceptable.

Focusing on viewer view trend analysis over time and expanding on the previous example shown in Figure 22 for time (T-State = 1), the following OLFVmv GOP chunk priorities are assigned for the independent DVB and P2P channels as follows, where V3 for T-State =1 is the most prevalent left core camera view and V4 is the most prevalent right core camera view:

<u>**T-State = 1**</u>

Key:

V_{DVB} Priority Level = BLUE
V_{P2P} Priority Level = GREEN

Priority:

1= Most Important
24 = Least Important

Core Camera View →	**V1**	**V2**	**V3**	**V4**	**V5**	**V6**	**V7**	**V8**
Base Layer	**13**	**7**	**1**	**2**	**10**	**16**	**19**	**22**
Metadata Layer	**14**	**8**	**3**	**4**	**11**	**17**	**20**	**23**
Enhanced Layer	**15**	**9**	**5**	**6**	**12**	**18**	**21**	**24**

Figure 24: Example of OLFVmv video content distribution between the DVB and P2P channels at T-State = 1

As shown in Figure 24 above, the DVB channel is prioritized to send the most prevalently viewed left and right core camera views V3 and V4. By design, the dual DVB channels have enough capacity using MPEG SVC . Thus the priority of the HDTV video content for the most prevalently viewed left and right core camera views are set to DVB channel Priority = 1 and 2 respectively for the base layer, and descending priorities for the other sub-layers through priority level 6 as shown above.

While a wide variety of priority schemes can be used to prioritize the P2P channel video content (*see* Section 5.2, Future Work), the above example uses a method that assigns the highest P2P channel priority to left and right core camera views that are left and right adjacent to the most prevalently viewed left and right DVB core camera views. Thus the P2P channel Priorities = 7, 8, 9, and 10, 11, 12 are set for camera views V2 and V5 base layer, metadata layer and enhanced layer video content respectively. The algorithm then increments outward to the left and to the right with descending priorities. For instance, in this example, the V1 and V6 base layer, then the metadata layer and then the enhanced layer video content is assigned P2P channel Priority = 13, 14, 15, and 16, 17, 18 respectively, and so.

An exemplary sample algorithm to set the initial priorities for the DVB (priorities 1-6) and the P2P channel (priorities 7-24) based on the viewing histogram is provided below:

Continued from Figure 19, above[5].

```
%% Set priorities masks for OLFVmv.  Each row is for a different layer: 1 = Base, 2 =
Metadata, 3 = Enhanced

% for trend camera 1->2 or fliplr (flip left-to-right) for camera 7 <-8
global OLFVMV_Channel_Priorities_Mask
OLFVMV_Channel_Priorities_Mask (:,:,1)= ...
  [1 2 7 10 13 16 19 22;
   3 4 8 11 14 17 20 23;
   5 6 9 12 15 18 21 24];

% for trend camera 2->3 or fliplr (flip left-to-right) for camera 6 <-7
OLFVMV_Channel_Priorities_Mask (:,:,2)= ...
   [7 1 2 10 13 16 19 22;
    8 3 4 11 14 17 20 23;
    9 5 6 12 15 18 21 24];

% for trend camera 3->4 or fliplr (flip left-to-right) for camera 5 <-6
OLFVMV_Channel_Priorities_Mask (:,:,3)= ...
   [13 7 1 2 10 16 19 22;
    14 8 3 4 11 17 20 23;
    15 9 5 6 12 18 21 24];

% for trend camera 4->5 or fliplr (flip left-to-right) for camera 4 <-5
OLFVMV_Channel_Priorities_Mask (:,:,4)= ...
   [19 13 7 1 2 10 16 22;
    20 14 8 3 4 11 17 23;
    21 15 9 5 6 12 18 24];

% for trend camera 5->6 or fliplr (flip left-to-right) for camera 4 <-5
OLFVMV_Channel_Priorities_Mask (:,:,5)= ...
   [22 19 13 7 1 2 10 16;
    23 20 14 8 3 4 11 17;
    24 21 15 9 5 6 12 18];

OLFVMV_Channel_Priorities_Mask (:,:,6)= ...
   [22 19 16 13 7 1 2 10;
```

[5] See Appendix C for a complete list and explanation of variables.

```
    23 20 17 14 8 3 4 11;
    24 21 18 15 9 5 6 12];

OLFVMV_Channel_Priorities_Mask (:,:,7)= ...
   [22 19 16 13 10 7 1 2 ;
    23 20 17 14 11 8 3 4 ;
    24 21 18 15 12 9 5 6];

OLFVMV_Channel_Priorities_Mask (:,:,8)= ...
   [22 19 16 13 10 7 2 1 ;
    23 20 17 14 11 8 4 3 ;
    24 21 18 15 12 9 6 5];

%% Algorithm to initialize DVB and P2P channel priorities for OLFVmv at T=1

function set_OLFVMV_channel_priorities_init

global Type_Index_OLFVMV
global Channel_Priorities
global Channel_Histogram
global Num_Cameras
global OLFVMV_Channel_Priorities_Mask

% Priorities for each camera, 1 = highest P2P priority,
% n = Num_Layers*Num_Cameras is the lowest P2P priority

% Determine what highest count is in histogram and for what camera number
% it occurs at, where each row number = the camera bin, e.g. Camera 1 = row
% 1, and so on.

Histogram_Time_Index = 1;            % This is for T state T=1 initialization only

Max_Histogram_Camera_Count = max(Channel_Histogram(:,Histogram_Time_Index));

for Max_Histogram_Camera_Index = 1:Num_Cameras
                              % Find the first occurrence where
                              % the max bin count occurs and

   if Max_Histogram_Camera_Count ==
Channel_Histogram(Max_Histogram_Camera_Index,Histogram_Time_Index)
                              % break to preserve the index
      break

   end % end - if

end % end - for Max_Index = 1:Num_Cameras

% Test to see the max camera is at the far left (==1) or right(==Num_Cameras)
% and if so, set the mask tending from the far left or right respectively
if (Max_Histogram_Camera_Index == 1) || ...
      (Max_Histogram_Camera_Index == Num_Cameras)

   Channel_Priorities(:,:,Histogram_Time_Index,Type_Index_OLFVmv) = ...
      OLFVMV_Channel_Priorities_Mask (:,:,Max_Histogram_Camera_Index);

% Otherwise, must be cameras 2 through Num_Cameras-1, so determine if the
```

```
% adjacent camera to the left or right is the next highest in the histogram.
% If adjacent cameras counts are equal, default to a right trend

% Test if next highest count camera is to the right or equal
elseif (Channel_Histogram(Max_Histogram_Camera_Index-1,Histogram_Time_Index) <= ...
       Channel_Histogram(Max_Histogram_Camera_Index+1,Histogram_Time_Index))

% If so, set the mask
    Temp_Mask_Index = Max_Histogram_Camera_Index;
    Channel_Priorities(:,:,Histogram_Time_Index,Type_Index_OLFVmv) = ...
       OLFVMV_Channel_Priorities_Mask (:,:,Temp_Mask_Index);

% Otherwise the trend must be to the left so flip the mask over so the
% priorities go toward the left, using a transposed index of the mask, e.g.
% N = 1 -> Num_Cameras, N= 2 -> Num_Cameras -1, thus Index =
% Num_Cameras - N + 1

else
   Temp_Mask_Index = Num_Cameras - Max_Histogram_Camera_Index + 1;
   Channel_Priorities(:,:,Histogram_Time_Index,Type_Index_OLFVmv) = ...
       fliplr(OLFVMV_Channel_Priorities_Mask(:,:,Temp_Mask_Index));

end % end - if

fprintf('execution complete: set_OLFVMV_channel_priorities_init \n')

end % end - set_OLFVMV_channel_priorities_init
```

Figure 25: Algorithm to initialize P2P channel priorities at T-State = 1

3.4.5.2 Trend Video Content Prioritization

The second part of the video content prioritization method it to alter the video content prioritization based on viewer trend analysis over time. To expand on the above example, assume that in T-State = 2, the viewing trend of the most prevalently viewed left and right core camera views shifts from V3 and V4 to V4 and V5 as follows:

T-State = 2

Key:

V_{DVB} = BLUE	
V_{P2P} = GREEN	

Priority:
1= Most Important
12 = Least Important

Viewing trend from T-State =1 to T-State = 2 (Left core view)

Viewing trend from T-State =1 to T-State = 2 (Right core view)

Core Camera View →	V1	V2	V3	V4	V5	V6	V7	V8
Base Layer	19	13	7	1	2	10	16	22
Metadata Layer	20	14	8	3	4	11	17	23
Enhanced Layer	21	15	9	5	6	12	18	24

→ Corresponding Viewing Trend Vector

Figure 26: Example of video content distribution between the DVB and P2P channels from T-State =1 to T-State = 2

For the DVB video content, as before, the two most prevalent left and right core camera views (V4 and V5 in T-State = 2) are set to Priority = 1 and 2 respectively for the base layer, and then descending priorities for the other sub-layers through priority level 6 as shown above.

For the P2P content, based on the fact that the viewing trend is pointing to the right (e.g., a soccer ball is kicked to the right of the field) a right pointing trend vector is established. As a result, the priorities are shifted to the right; the most prevalent (adjacent) left and right views (V3 and V4) are given descending priority for the P2P channel starting with the base and metadata layers, followed by the enhancement layer. The priorities continue to descend, followed by the next left and right adjacent core camera view's (V2 and V6) base and metadata layers, followed by the enhanced layer as shown above in Figure 26.

An exemplary algorithm to establish the viewing trend vector and set the corresponding priorities for the P2P channel is as follow:

Continued from Figure 25, above[6].

```
%% Figure 27 from Thesis for  - Set DVB and P2P channel priorities for OLFVmv

function set_OLFVMV_channel_priorities

global Channel_Priorities
global Channel_Histogram
global Num_Cameras
global OLFVMV_Channel_Priorities_Mask
global Num_Sim_Run_Time_Ticks
global Type_Index_OLFVmv

% Priorities for each camera, lowest number = highest P2P priority,
% highes number n = Num_Layers*Num_Cameras is the lowest P2P priority

% Starting with T=2 and for each time tick after, take the histogram results
% from the last T-State, compare them to the current T-State, determine if the
% trend is to the left or right and set the priority mask accordingly

Max_Histogram_Camera_Index_Last = 1;          % Initialized the previous T-State as 1

for Histogram_Time_Index = 2:Num_Sim_Run_Time_Ticks

   % Determine what highest count is in histogram for this T-State and
   % what camera number each occurred at.  The function find returns row
   % and column so this needs to be reduced to just column.

   Max_Histogram_Camera_Count_Current =
max(Channel_Histogram(:,Histogram_Time_Index));
   for Max_Histogram_Camera_Index_Current = 1:Num_Cameras
                              % Find the first occurrence where
                              % the max bin count occurs and

      if Max_Histogram_Camera_Count_Current == ...
             Channel_Histogram(Max_Histogram_Camera_Index_Current,Histogram_Time_Index)
                              % break to preserve the index
         break

      end % end - if

   end % end - for Max_Histogram_Camera_Index_Current = 1:Num_Cameras

   % Test to see the max camera is at either end, and if so, set the mask
```

6 See Appendix C for a complete list and explanation of variables.

```
    % trending from the end

    if (Max_Histogram_Camera_Index_Current == 1)||...
          (Max_Histogram_Camera_Index_Current == Num_Cameras)

       Channel_Priorities(:,:,Histogram_Time_Index,Type_Index_OLFVmv) = ...
          OLFVMV_Channel_Priorities_Mask(:,:,Max_Histogram_Camera_Index_Current);

    % Otherwise, must be cameras 2 through Num_Cameras-1, so determine if the
    % trend is from the left to right.  Default is to the right.

    % Compare where the current max camera histogram point is the current
    % T-State compared to where it was in the last T-State.  If the current
    % T-State index is greater, then the trend is to the right

    elseif (Max_Histogram_Camera_Index_Current >= Max_Histogram_Camera_Index_Last)

     % If so, set the mask
        Temp_Mask_Index = Max_Histogram_Camera_Index_Current;
        Channel_Priorities(:,:,Histogram_Time_Index,Type_Index_OLFVmv) = ...
          OLFVMV_Channel_Priorities_Mask(:,:,Temp_Mask_Index);

% Otherwise the trend must be to the left so flip the mask over so the
% priorities go toward the left, using a transposed index of the mask, e.g.
% N = 1 -> Num_Cameras, N= 2 -> Num_Cameras -1, thus Index =
% Num_Cameras - N + 1

    else
       Temp_Mask_Index = Num_Cameras - Max_Histogram_Camera_Index_Current + 1;
       Channel_Priorities(:,:,Histogram_Time_Index,Type_Index_OLFVmv) = ...
          fliplr(OLFVMV_Channel_Priorities_Mask (:,:,Temp_Mask_Index));

    end % end - if

% now that we are done testing, remember the index where the max camera
% occurred for the next loop.  Thus _Current becomes _Last.

Max_Histogram_Camera_Index_Last = Max_Histogram_Camera_Index_Current;

end % end - for Histogram_Time_Index = 2:Num_Sim_Run_Time_Ticks

fprintf('execution complete: set_OLFVMV_channel_priorities \n');

end % end - function set_OLFVMV_channel_priorities
```

Figure 27: Algorithm to set P2P channel priorities at T-State = 2 based on viewing directional trends

3.4.6 P2P-MS Hierarchical Network Coding and Network Management of the P2P Selected Network Video Content

As a final step in the implementation of OLFVmv, utilizing the above defined P2P video content priorities, an enhanced version of Hierarchical Network Coding (HNC) as described in the paper "Video Streaming with Network Coding" [NguyenNguyenCheung2010] is proposed. To describe the proposed modification to HNC, it is prudent to first provide an overview of HNC overall.

3.4.6.1 Overview of Hierarchical Network Coding

The paper "Video Streaming with Network Coding" [NguyenNguyenCheung2010] discloses, and OLFVmv, adopts a video Content Distribution Network (CDN) using a TCP-IP protocol (*id.* at pg. 9). The network topology is further made up of "source" and "intermediate nodes" (*id.*) as is further shown below:

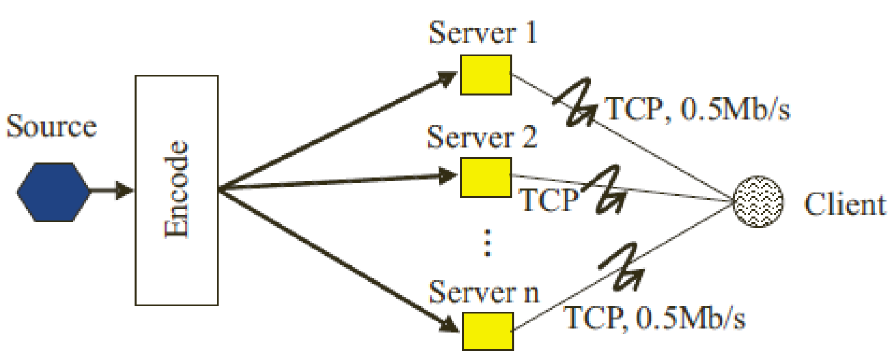

Figure 28: HNC network path diversity streaming topology showing source and intermediate nodes
([NguyenNguyenCheung2010] at Fig. 1, pg. 9).

Further, based on the importance of data, HNC correlates the importance of video content to the redundancy level that the packets are transmitted over the network, as further shown below:

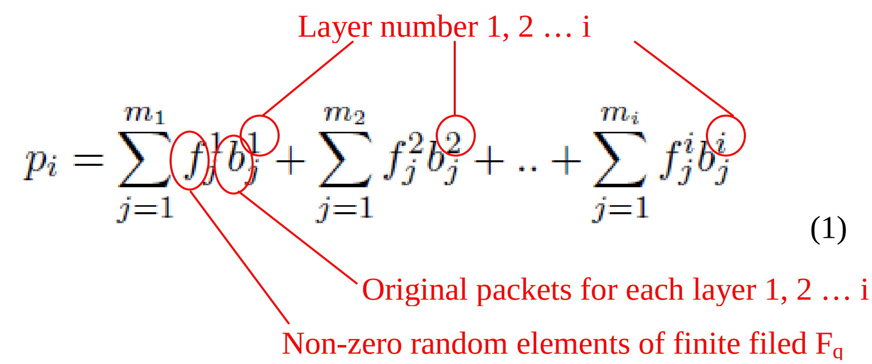

Figure 29: HNC network packet encoding
([NguyenNguyenCheung2010] at Equation (1), pg. 12).

Where as stated in the paper "Video Streaming with Network Coding" [NguyenNguyenCheung2010]:

$b_1^i, b_2^i, \ldots\ b_m^i$. denotes all packets within a chunk.

b_j^i represents the original packet for each layer 1, 2 … i.

f_j^i are non-zero random elements of finite filed F_q.

p_i represents the aggregated packet information for all layers.

An exemplary implementation is shown below, where a1 and a2 are packets for the most important base layer and b1 and b2 are lower priority, thus b1 and b2 are less redundant packets for the enhancement layer.

COMPARE CODING SCHEMES WITH 2 LAYERS DATA

Highest priority (thus most redundant) e.g, base layer content or most important core camera views

Lower priority (thus least redundant) e.g., enhancement layer content or least important core camera views

Uncoded	WLNC	Hierarchical NC	RNC
a_1	a_1	a_1	a_1
a_2	a_2	a_2	a_2
	$a_1 + a_2$	$a_1 + a_2$	$a_1 + a_2$
b_1	b_1	$a_1 + b_1$	$a_1 + b_1$
b_2	b_2	$a_1 + b_2$	$a_1 + b_2$
	$b_1 + b_2$	$a_1 + b_1 + b_2$	$a_1 + b_1 + b_2$
		$a_2 + b_1$	$a_2 + b_1$
		$a_2 + b_2$	$a_2 + b_2$
		$a_2 + b_1 + b_2$	$a_2 + b_1 + b_2$
		$a_1 + a_2 + b_1$	$a_1 + a_2 + b_1$
		$a_1 + a_2 + b_2$	$a_1 + a_2 + b_2$
		$a_1 + a_2 + b_1 + b_2$	$a_1 + a_2 + b_1 + b_2$
			b_1
			b_2
			$b_1 + b_2$

Figure 30: Hierarchical Network Coding (HNC) video content layer redundancy based on priority

([NguyenNguyenCheung2010] at Table 1, pg. 13).

3.4.6.2 Application of Hierarchical Network Coding to FTV

While the focus of HNC as disclosed in the paper "Video Streaming with Network Coding" [NguyenNguyenCheung2010] is to assign redundancy of packets to be correlated to the prioritization of video content layers within a single video stream, *OLFVmv anticipates using the same methods to assign redundancy of packets to be* *correlated for <u>both</u> the priorities of: (1) adjacent core camera view metadata and base layer content, and (2) separately the enhancement layers of the adjacent core camera view video content as shown in Figure 26, above.*

As a result, the higher priority core camera video content will be more redundantly transmitted over the P2P channel, thus increasing availability of the most likely requested (e.g., trending) video content, versus the unintelligent methods employed by DIOMEDES. Employing a modification to the equation shown in Figure 29 for HNC within a single video channel, an improved HNC method for transporting OLFVmv using multiple video channels is as follows:

$$pi = \sum_{n=1}^{MAX\_CORE\_CAMERAS} \left(\sum_{j=1}^{m\_META} f_{n,j}^{META}\, b_{n,j}^{META} + \sum_{j=1}^{m\_BASE} f_{n,j}^{BASE}\, b_{n,j}^{BASE} + \sum_{j=1}^{m\_ENHANCED} f_{n,j}^{ENHANCED}\, b_{n,j}^{ENHANCED} \right)$$

Figure 31: Expression of OLFVmv P2P packets within chunks

Where:

$b_{n,1}^{META}$, $b_{n,2}^{META}$, ... b_{n,m_META}^{META}. denotes all packets within a chunk, for example the META data layer for each core camera n.

$b_{n,j}^{META}, b_{n,j}^{BASE}, b_{n,j}^{ENHANCED}$ represents the original packet for the META, BASE and ENHANCED layers for each core camera n.

$f_{n,j}^{META}, f_{n,j}^{BASE}, f_{n,j}^{ENHANCED}$ are non-zero random elements of finite filed F_q for each core camera n.

p_i represents the aggregated packet information for all layers.

As a result, turning to the example shown in Figure 26, the HNC encoded packet content would look like the following:

Table 2: Hierarchical Network Coding (HNC) packet classes as applied to OLFVmv core camera video content distribution over the P2P channel using the priority map as shown in Figure 26

Packet Class (n)	Packet Class Probability (Pn)	Hierarchical NC for FTV Video Content
1	P_1	$V3_{Base}$ $V3_{Meta}$ $V3_{Enhanced}$ $V3_{Base} + V3_{Meta}$ $V3_{Base} + V3_{Enhanced}$ $V3_{Meta} + V3_{Enhanced}$ $V3_{Base} + V3_{Meta} + V3_{Enhanced}$
2	P_2	$V3_{Base} + V6_{Base}$ $V3_{Base} + V6_{Meta}$ … $V3_{Meta} + V6_{Base}$ $V3_{Meta} + V6_{Meta}$ … $V3_{Enhanced} + V6_{Base}$ … $V3_{Base} + V3_{Meta} + V6_{Base}$ … $V3_{Base} + V3_{Meta} + V3_{Enhanced} + V6_{Base} + V6_{Meta} + V6_{Enhanced}$
…	…	…
12	P_6	… $V3_{Base} + V3_{Meta} + V3_{Enhanced} + V6_{Base} + V6_{Meta} + V6_{Enhanced} + V2_{Base} + V2_{Meta} + V2_{Enhanced} + V7_{Base} + V7_{Meta} + V7_{Enhanced} + V1_{Base} + V1_{Meta} + V1_{Enhanced} + V8_{Base} + V8_{Meta} + V8_{Enhanced}$
SUM	**= 1**	

From this, the P2P channel's TS Chunker then assigns a probability P_n, to each packet class *c*, shown above. After the packet class is chosen based on probability P_n, a packet is randomly and uniformly chosen from a given packet class and generated as is the case for HNC.

For OLFVmv, the calculation of a given probability *P* for a given class c, shall be as follows, however as discussed in Section 5.2 - Future Work, additional probability models can be developed and would be beneficial to improved system performance.

$$Pn = \frac{\frac{C!}{(n/(C!-n))}}{\sum_{n=1}^{C}\frac{C!}{(n/(C!-n))}}$$

Figure 32: Expression of P2P packet priorities within chunks

Where:

n represents the packet class number for a given packet class.

Pn is the probability weighting factor for a given packet class number.

C is the total number of packet classes.

From the above, and turning to the example shown in Table 2, where C=6, the probabilities for each packet class are as follows:

Table 3: Example - Hierarchical Network Coding (HNC) packet class priorities as applied to FTV core camera video content distribution over the P2P channel

Packet Class (n)	Packet Class Probability (Pn)
1	0.407339
2	0.203953
3	0.136158
4	0.102261
5	0.081923
6	0.068365
SUM	**1.000000**

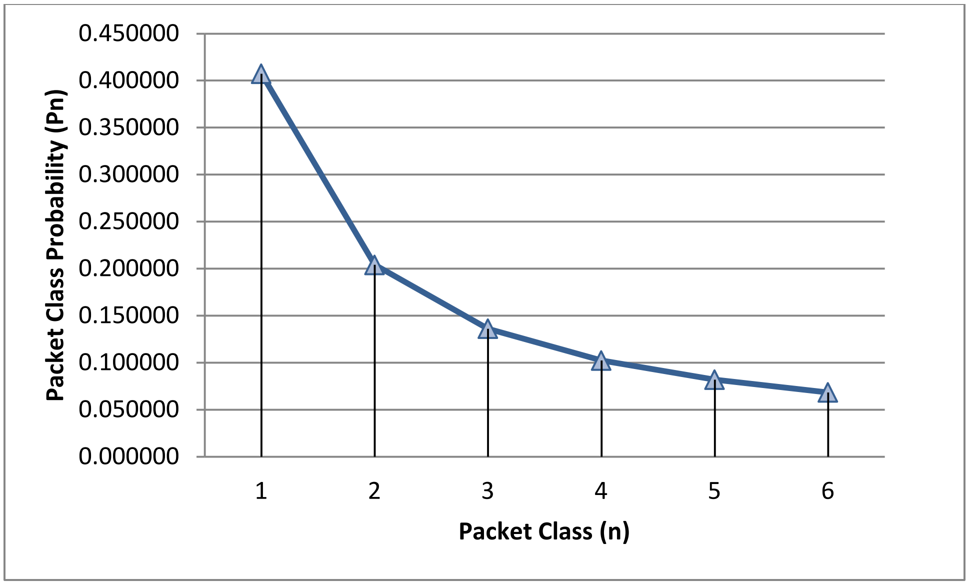

Figure 33: Example - Hierarchical Network Coding (HNC) packet class priorities as applied to OLFVmv core camera video content distribution over the P2P channel

While the focus of HNC as disclosed in the paper "Video Streaming with Network Coding" [Nguyen_Nguyen_Cheung2010] is to assign redundancy of packets to be correlated to the prioritization of video content layers within a single video stream, this thesis anticipates using the same methods to assign redundancy of packets to be correlated for both the priorities of (1) adjacent core camera views and (2) the content layers within the core camera views as shown in Figure 26, above. By doing so, additional probability distribution schemes can be built, for example: to separate the less important enhanced layer into a separate lower priority class.

Chapter 4: Simulation / Testbed Results

4.1 Simulation Approach Overview

In this section the performance of the proposed methods of OLFVmv are fully evaluated. The motivation is to test the performance of the DIOMEDES content packet availability as compared to OLFVmv content packet availability. Accordingly, both the DIOMEDES and OLFVmv schemes were simulated using the same simulated video data chunk rate, over a spectrum of viewing patterns (e.g., such as both random and oscillating left to right viewing patterns) and over a spectrum of bandwidth conditions.

4.2 Simulation Setup and Assumptions

There are a number of simulation inputs and assumptions that have been modeled, including:

1) Viewer behavior
2) Network bandwidth conditions
3) Video bandwidth requirements
4) DVB content distribution assumptions

Subsequently, each of these topics are described below in more detail.

4.2.1 Viewer Behavior Assumptions

To simulate a suitable population, N =100 users were assumed for all simulations. Further, simulations over a wide variety of viewer conditions have been modeled.

4.2.1.1 Varying Viewer View Gaussian Distribution

First, the simulations were run with viewer's viewing patterns being randomized around a mean viewing position (whereas the mean viewing position is described below). For the randomization, each user was assigned a normalized Gaussian randomly distributed viewing offset (to the left or right) of a mean viewing position. In the real-world, this simulates the fact that for example, during an event, most of the viewers will be looking at a specific activity location (say the location of the soccer ball during a soccer game) plus or minus some viewer to viewer variation.

For the Gaussian distribution, iterative simulations were run with varying the Sigma of the normal distributions as follows:

Iterative Simulations were run with Sigma = [0.1, 0.5, 1.0, 2.0]

4.2.1.2 Varying Mean Viewer View Positions

Second, the simulations were run iteratively with varying the viewing pattern mean by using a saw-tooth oscillation pattern that goes back and forth (far left to far right and then back again) at varying time period rates (τ).

Iterative Simulations were run with Viewer Oscillation Period (τ)

= [5, 10, 50] seconds

4.2.2 Network Bandwidth Conditions

Because OLFVmv optimizes P2P bandwidth utilization, the simulations utilized bandwidth rates on the lower end of the bandwidth spectrum. In order to test the limitations of OLFVmv as compared to DIOMEDES, the simulations for OLFVmv started at 2 Mbps. Performance of OLFVmv and DIOMEDES were also simulated at higher bandwidths for the P2P channel including 16, 32, and 64 Mbps.

Iterative Simulations were run with a given Viewer's P2P bandwidth

= [2, 16, 32, 64] Mbps

Additionally, each user was assigned a Gaussian distributed “impairment” (both positive and negative) via a normal distribution with a sigma = 0.1.

4.2.3 Video Content Bandwidth Requirements and Assumptions

Related to the video content bandwidth requirements, the following assumptions have been applied to the simulation model:

1) The assumed bandwidth required for a full HDTV 1080p video channel, including the base, metadata and enhanced layers is 19.3 Mbps [VideoBandwidthEstimates]

2) The base layer video is assumed to be SDTV 480i, which requires 6 Mbps of bandwidth (a standard cable TV channel)

3) Thus calculating from above, the enhanced layer bandwidth = 19.3 – 6 ~ 13.3 Mbps.

4) The assumed metadata layer bandwidth including 3-D depth maps = 2 Mbps

5) Because MVV generates video frames for a given P2P transmitted view using B and P frames based an adjacent primary core camera view sent via the DVB channel, the adjacent P2P views (see Figure 20 above) use less bandwidth because there is no requirement for the P2P channel to transmit I frames.

6) From this, the assumed video compression rates for P2P video content (or an adjacent DVB channel next to a primary DVB channel) is as follows as compared to an I-Frame [InterFrameCompression] from a primary core camera view transmitted via the DVB channel:

   B-Frame size ~ 25% I-Frame size

   P-Frame size ~ 50% of an I-Frame size

7) Assume an adjacent P2P channel for a HDTV view would be a B-Frame and P-Frame based off of the adjacent channel consisting of nine B-Frames followed by one P-Frame, thus B B B B B B B B B P B B B…

Thus using the above and calculating as needed, yields:

For a primary channel (e.g., left most DVB channel)

- Primary DVB Base (B) layer = 6 Mbps
- Primary DVB Enhanced (E) layer = 13.3 Mbps → 12 Mbps
- Primary DVB Metadata (M) layer = 2 Mbps

Total Primary B+E+M ~ 21.2 Mbps → 20 Mbps

For an adjacent channel (e.g., any P2P or DVB channel other than the primary DVB channel) to following calculations apply:

- Adjacent Base (B) layer = 6 x (0.25x9+0.5x1)/10 = 1.65 Mbps → 2 Mbps
- Adjacent Metadata (M) layer = 2 Mbps
- Adjacent Enhanced (E) layer = 13.3 x (0.25x9+0.5x1)/10 = 3.66 Mbps → 4 Mbps

Total Primary B+E+M ~ 7.3 Mbps → 8 Mbps

Table 4: Video Content Bandwidth Requirements and Assumptions.

Description	***Bandwidth Used per Channel (Mbits/s)***
Video Content Bandwidth Requirements	From above: For a primary channel (e.g., left most DVB channel) • Primary Base (B) layer = 6 Mbps • Primary Enhanced (E) layer = 13.3 Mbps → 12 Mbps • Primary Metadata (M) layer = 2 Mbps **Total Primary B+E+M ~ 21.2 Mbps → 20 Mbps** For an adjacent channel (e.g., any P2P or DVB channel other than the primary channel) • Adjacent Base (B) layer = 6 x (0.25x9+0.5x1)/10 = 1.65 Mbps → 2 Mbps • Adjacent Enhanced (E) layer = 13.3 x (0.25x9+0.5x1)/10 = 3.66 Mbps → 4 Mbps • Adjacent Metadata (M) layer = 2 Mbps **Total Primary B+E+M ~ 7.3 Mbps → 8 Mbps**
Channel Capacities	
DVB Channel Throughput (assume DVB-S, ATSC)	Assume 28 Mbps (e.g., 1 primary HDTV 3D channel (B+E+M) plus 1 adjacent HDTV 3D channel (B+E+M)
P2P Channel Throughput (assume no P2P or Network coding)[7]	Mean_P2P_BW_Simulation_Rates = {2, 16, 32 and 64} Mbits/s, with Gaussian distribution of Sigma_P2P_BW = 0.1

4.2.4 DVB Content Distribution Assumptions

Based on the above calculations, both the OLFVmv and DIOMEDES DVB channel capacity is assumed to provide 28 Mbps.

Turning to Figure 34 and Figure 35, shown below for OLFVmv, assuming that the primary core camera views V4 and V5 utilize priorities 1 through 6 for each of the base, metadata and enhanced layers as shown in Figure 34, then the sum total of

[7] Any gains from P2P packets and gains from network coding are assume to be the same between DIOMEDES and OLFVmv for the sake of bandwidth throughput comparisons.

priorities 1 through 6 = 28 Mbps (shown in BLUE in Figure 35). Thus sufficient throughput is assumed to exist for all of the V4 and V5 layers to allow transmission over the DVB channel. As the histogram of what viewers are watching changes, the core views transmitted by the DVB channel changes, thus the below is simply an example.

Key:

DVB Channel Priority / Bandwidth
P2P Channel Priority / Bandwidth

Core Camera View →	V1	V2	V3	V4	V5	V6	V7	V8
Base Layer	19	13	7	1	2	10	16	22
Metadata Layer	20	14	8	3	4	11	17	23
Enhanced Layer	21	15	9	5	6	12	18	24

Figure 34: Example OLFVmv Priority Matrix

Core Camera View →	V1	V2	V3	V4	V5	V6	V7	V8
Base Layer	2Mbps	2Mbps	2Mbps	6Mbps	2Mbps	2Mbps	2Mbps	2Mbps
Metadata Layer	2Mbps	2Mbps	2Mbps	2Mbps	2Mbps	2Mbps	2Mbps	2Mbps
Enhanced Layer	4Mbps	4Mbps	4Mbps	12Mbps	4Mbps	4Mbps	4Mbps	4Mbps
Total	8Mbps	8Mbps	8Mbps	20Mbps	8Mbps	8Mbps	8Mbps	8Mbps

Figure 35: OLFVmv DVB Transport Bandwidth used for Figure 34

Turning now to Figure 36 and Figure 37, shown below for DIOMEDES, given that DIOMEDES ranks the V1 base layer as priority = 1, transmits three core camera views, and ranks metadata and then the enhanced layer of the three core camera views in descending priorities; this yields the priorities shown in Figure 36. Using these priorities, the allocation of the DVB channel capacity of 26 Mbps is shown in BLUE in Figure 37.

Core Camera View →	V1	V2	V3	V4	V5	V6	V7	V8
Base Layer	1	3	5	10	13	16	19	22
Metadata Layer	2	4	6	11	14	17	20	23
Enhanced Layer	7	8	9	12	15	18	21	24

Figure 36: DIOMEDES Priority Matrix

Core Camera View →	V1	V2	V3	V4	V5	V6	V7	V8
Base Layer	6Mbps	2Mbps	2Mbps	2Mbps	2Mbps	2Mbps	2Mbps	2Mbps
Metadata Layer	2Mbps	2Mbps	2Mbps	2Mbps	2Mbps	2Mbps	2Mbps	2Mbps
Enhanced Layer	12Mbps	4Mbps	4Mbps	4Mbps	4Mbps	4Mbps	4Mbps	4Mbps
Total	20Mbps	8Mbps	8Mbps	8Mbps	8Mbps	8Mbps	8Mbps	8Mbps

Figure 37: DIOMEDES DVB Transport Bandwidth used for Figure 36

4.2 Results

The simulation results produced an output of the probabilistic likelihood that a given viewer's receiver would contain the desired core camera video content so that the receiver could render the viewer's desired synthesized virtual view at position Vn. Each of the scenarios of viewing pattern oscillation rate, viewer random offset distribution, and randomized per-viewer available bandwidth were tested. The full results are provided in Appendix C and summarized in this section.

Overall, the results reflect the probability of a "hit" (e.g., a "hit" is "1" for a given viewer if the desired content for view Vn was available because the Vn_Left and Vn_Right views were available). Suffice to say, a hit for a given viewer was "0" if either the Vn_Left or Vn_Right core camera views were missing, thereby negating the ability to produce a synthesized view Vn that occurred between two core camera views.

For a given simulation scenario, the probability of a hit was calculated for the entire population of viewers and the probability of a hit was assigned a "P_Hit" (Probability of Hit) value for each of the layers (base, metadata, enhanced), and the combination of the layers (namely: base + metadata layers, base + enhanced layers, and base + metadata + enhanced layers)..

Thus as an example, the below plot shows that for a viewing pattern of 100 viewers looking side to side every 5 seconds (Oscillation Rate, "Osc" = 5 seconds) with a randomized view dispersion between viewers of Sigma = 1, the P_Hit rate for the OLFVmv of slightly less than 60% for a viewer to have the desired base layer (SDTV), metadata layer (MVC information) and enhanced layer (HDTV) content via a 2 Mbps P2P channel in combination with the DVB channel. Comparatively, under the same conditions, the likelihood of a viewer having the same content on a DIOMEDES system is less than 10%.

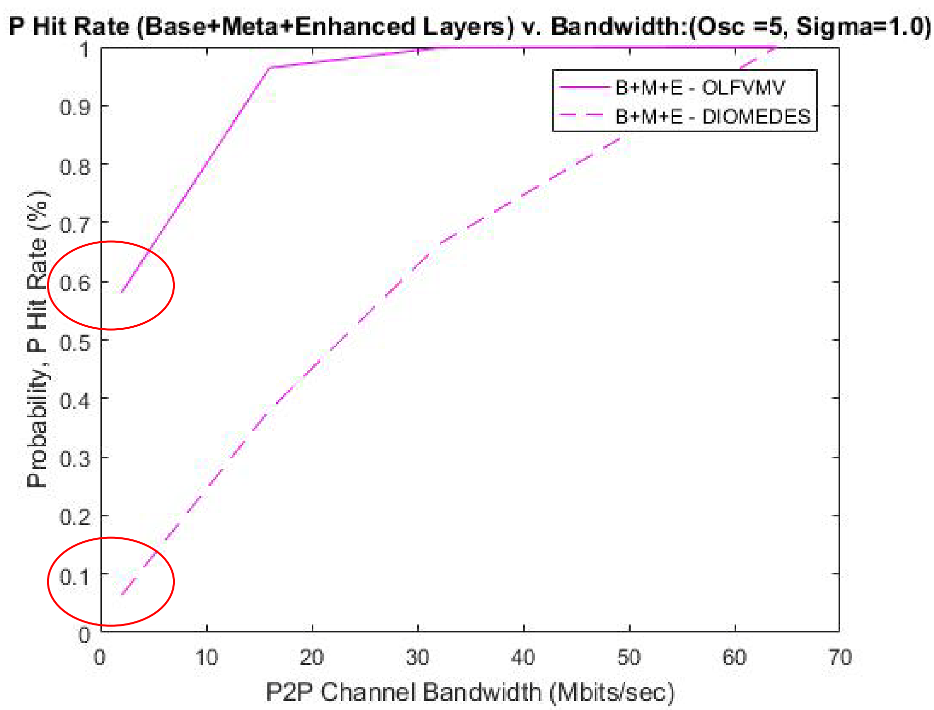

Figure 38: Example Simulation Results

Overall, OLFVmv outperformed DIOMEDES in all cases for P2P bandwidths of less than 64 Mbps and only at 64 Mbps did DIOMEDES' performance match that of OLFVmv. At a P2P channel bandwidth of 2 Mbps, a slow viewer oscillation rate (Osc = 50 seconds), and with a small Sigma (e.g., the dispersion of viewer's views among viewers was slight) OLFVmv outperformed DIOMEDES by 340%. For this case, because OLFVmv was able to easily anticipate the desired view based on the trend analysis, and thus adapt what content was sent - *nearly 98% of viewers had the desired base layer content available, while only 29% of DIOMEDES viewers had the desired based layer content available.* DIOMEDES was able to obtain parity only at a P2P channel bandwidth of 64 Mbps.

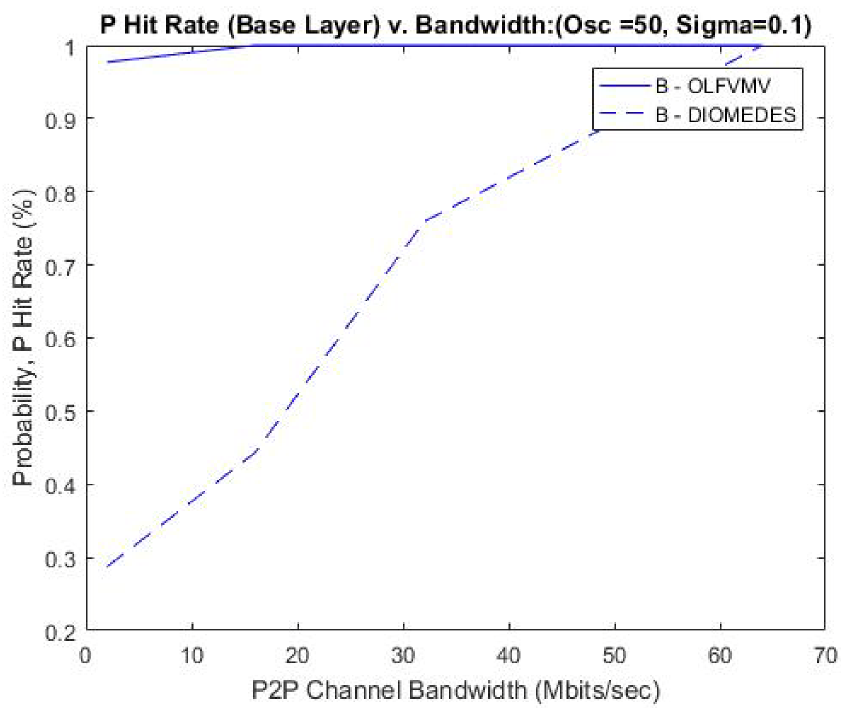

Figure 39: P Hit (Base Layer, Osc = 50, Sigma 0.1)

The trend tracked accordingly for the base plus metadata layers (Figure 40) and base plus metadata plus enhanced layers (Figure 41) where for OLFVmv the DVB channel was able to adapt to the desired view to send nearly 100% of the desired content to viewers via a 2 Mbps P2P channel. The corresponding P Hit rate for DIOMEDES was 0% via a 2 Mbps P2P channel based on the fact that DIOMEDES' priority allocation only allows one enhanced layer channel to be sent via the DVB channel, and the P2P channel at 2 Mbps is insufficient to transmit the adjacent core camera's enhanced layer data.

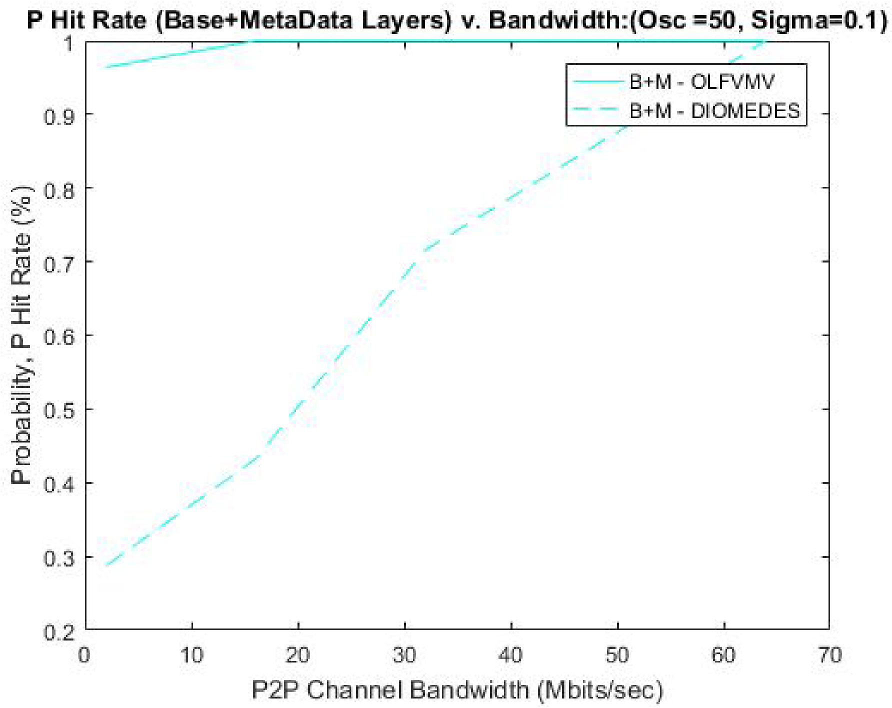

Figure 40: P Hit (Base + Metadata, Osc = 50, Sigma 0.1)

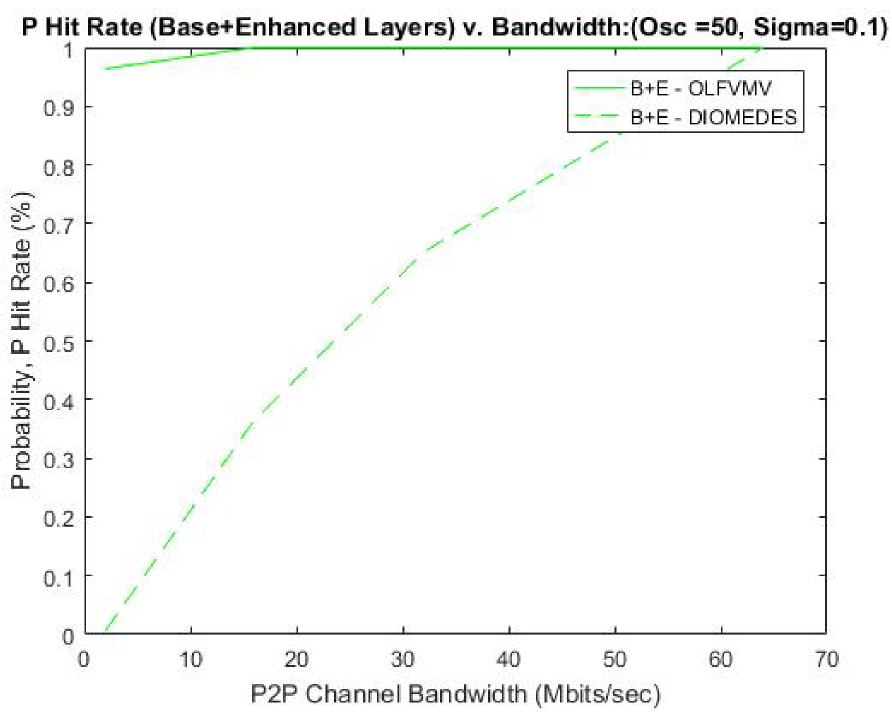

Figure 41: P Hit (Base + Enhanced Layers, Osc = 50, Sigma 0.1)

As a result of the DIOMEDES P2P channel at 2 Mbps being insufficient to transmit the Enhanced layer data, the P Hit rate for the base plus enhanced layer is likewise 0% as shown below.

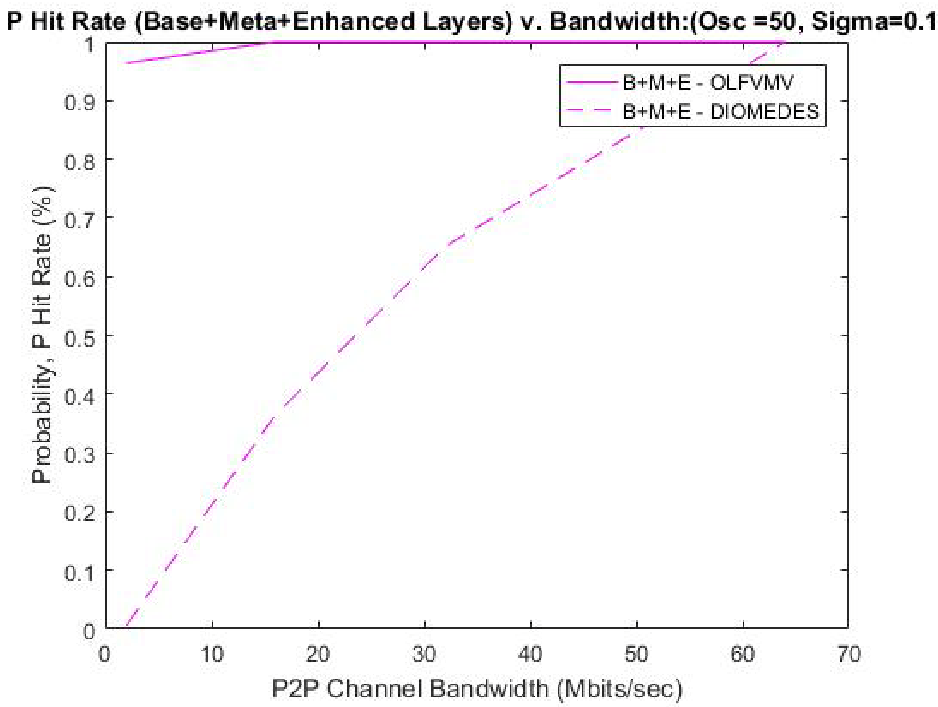

Figure 42: P Hit (Base + Metadata + Enhanced Layers, Osc = 50, Sigma 0.1)

In contrast to the largest disparities that occurred between OLFVmv and DIOMEDES as shown above (see Figure 39 through Figure 42), OLFVmv and DIOMEDES performance was closest to each other when the viewing pattern entailed a rapid oscillation (Osc = 5 seconds) and the randomized dispersion of viewpoints between viewer's was at the highest (Sigma = 2.0). As shown below in Figure 43 through Figure 46, OLFVmv far exceed the performance of DIOMEDES at lower P2P bandwidths throughputs. For example at a P2P bandwidth throughput of 2 Mbps, the performance of OLFVmv exceeded that of DIOMEDES by 190%, 179% and 454% for the base, base plus metadata and base plus metadata and enhanced layers respectively.

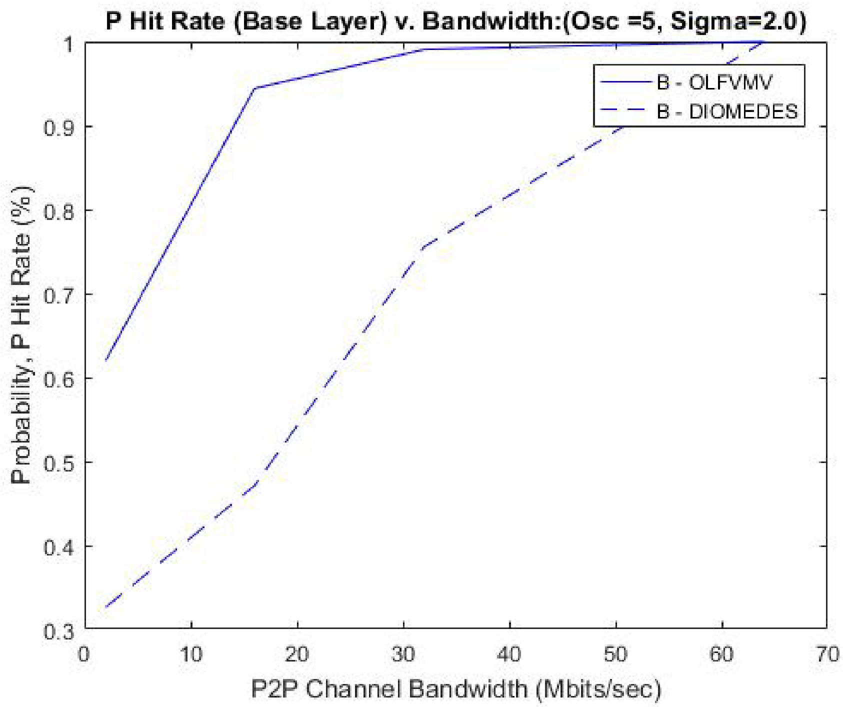

Figure 43: P Hit (Base Layer, Osc = 5, Sigma 2.0)

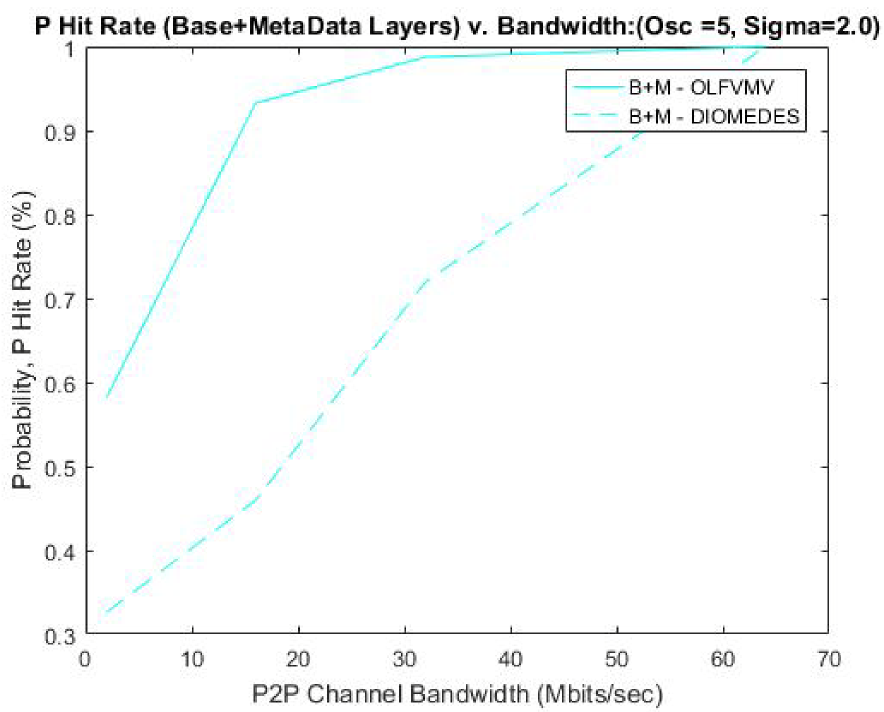

Figure 44: P Hit (Base plus Metadata Layers, Osc = 5, Sigma 2.0)

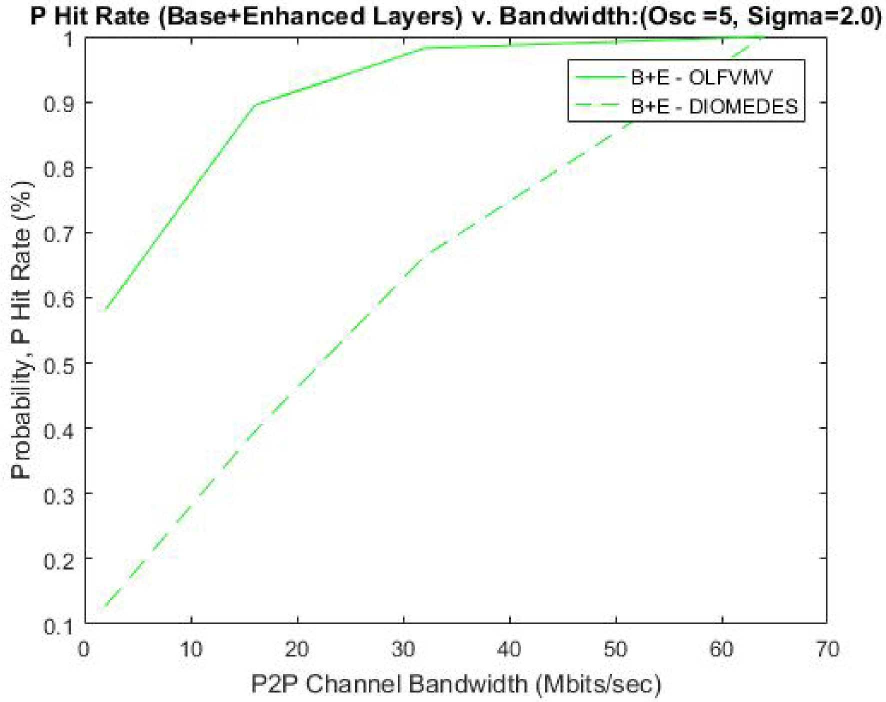

Figure 45: P Hit (Base + Enhanced Layers, Osc = 5, Sigma 2.0)

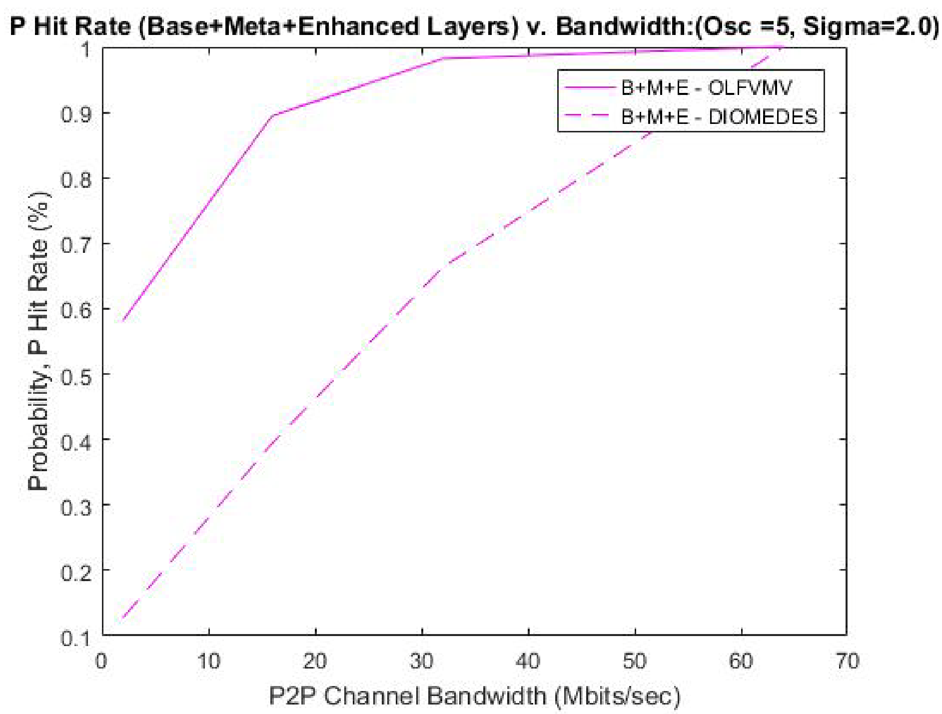

Figure 46: P Hit (Base + Metadata + Enhanced Layers, Osc = 5, Sigma 2.0)

Chapter 5: Conclusion and Future Work

5.1 Conclusion

Overall, related to the reliable transport of video content: OLFVmv's performance far exceeded the performance of DIOMEDES. The vast difference in OLFVmv's improved performance over DIOMEDES is based on OLFVmv's ability to adaptively sense and better prioritize video content. As a result, at low P2P system bandwidth throughputs such as 2 Mbps, OLFVmv outperformed DIOMEDES from a range of 190% (Sigma = 2, Osc = 5 for the base layer – see Figure 43 above) to infinitely better (Sigma = 0.1, Osc = 50 for the base plus metadata and enhanced layers – see Figure 41 above).

The positive performance of OLFVmv is important because it opens the door for the use of true live free viewpoint video using standard DVB channels augmented with a limited throughput P2P channel using 2 Mbps – 16 Mbps, to achieve the similar results of DIOMEDES at 64 Mbps.

5.2 Future Work

While Section 3.4.5 P2P-MS Network Video Content Selection Prediction, presents one possible algorithm for assigning priorities to core camera views and layers of video content, significant opportunities and research exists to enhance this work as follows:

1) First, as compared to Figure 24 and Figure 26, different prioritization models can be built and tested. For example, one option would be to place the enhanced layer of all the core cameras at a lower priority level as compared to the metadata layer and base layer priorities. Based on such changes, a tradeoff analysis can be made related to evaluating content availability for the

non-trending core camera views (which would increase as a result of higher priority base layer and metadata layer content) versus improved video quality as a result of the higher prioritization of the enhancement layer content.

2) Second, additional trend prediction models can be developed and tested, to optimize the prediction of future core video camera views.

3) Third, within the Hierarchical Network Coding algorithm, improved and adaptive probabilistic distribution models can be developed to assign redundancy to packets based on (1) the priority of the video content and (2) adaptation of priorities to compensate for network characteristics.

Bibliography

[Cheung2011] Cheung, et al., “Interactive Streaming of Stored Multiview Video Using Redundant Frame Structures”, *IEEE Transactions on Image Processing*, Vol. 20, No. 3, 744-761, March 2011.

[DIOMEDES D2.3 2012] DIOMEDES standard D2.3, “Final reference system architecture report”, Final Revision, dated Jan 31, 2012.

[DIOMEDES D3.6 2011] DIOMEDES standard D3.6, “Public report on 3D Audio / Video rendering and content adaption”, Final revision, dated 10/31/2011.

[DIOMEDES D4.4 2011] DIOMEDES standard D4.4, “Report on the developed audio and video codecs”, Final revision, Oct 31, 2011.

[DIOMEDES D4.5 2011] DIOMEDES standard D4.5, “Report on results of integrated MD-SMVD and P2P system”, Final revision, dated 12/31/2011.

[DIOMEDES_Flyer2013] “Distribution of MultiView Entertainment Using Content Aware Delivery Systems, 2013.

[Dufaux2013] Frederic Dufaux et al., “Emerging Technologies for 3D Video: Creation, Coding, Transmission and Rendering”, *Wiley Publishing*, 2013.

[ETSI EN300 421 V1.1.2:1997] ETSI EN 300 421 V1.1.2, “Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for 11/12 GHz satellite services”, August, 1997.

[ETSI EN300 429 V1.2.1:1998] ETSI EN 300 429 V1.2.1, “Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for cable systems”, April, 1998.

[ETSI EN302 304 V1.1.1:2004] ETSI EN 300 304 V1.1.1, “Digital Video Broadcasting (DVB); Transmission System for Handheld Terminals (DVB-H)”, November, 2004.

[ETSI EN302 307 V1.2.1:2009] ETSI EN 302 307 V1.2.1, “Digital Video Broadcasting (DVB); Second generation framing structure, channel coding and modulation systems for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications (DVB-S2)”, August, 2009.

[ETSI EN302 755 V1.2.1:2010] ETSI EN 302 755 V1.2.1, “Digital Video Broadcasting (DVB); Frame structure channel coding and modulation for a second generation digital terrestrial television broadcasting system (DVB-T2)”, October, 2010.

[ETSI TS102 034 V1.4.1:2009] ETSI TS 102 034 V1.4.1, “Digital Video Broadcasting (DVB); Transport of MPEG-2 TS Based DVB Services over IP Based Networks”, August, 2009.

[InterFrameCompression] Interframe compression estimates, at link: https://en.wikipedia.org/wiki/Inter_frame, last visited Nov. 13, 2016

[ISO/IEC 14496-10:2008] ISO/IEC 14496-10:2008, “Information technology -- Coding of audio-visual objects -- Part 10: Advanced Video Coding”, 2008.

[ISO/IEC 14496-10:2009] ISO/IEC 14496-10:2009, “Information technology -- Coding of audio-visual objects -- Part 10: Advanced Video Coding” extension, including the Multiview High profile, March 16, 2009.

[ISO/IEC 14496-10:2010] ISO/IEC 14496-10:2010, “Information technology -- Coding of audio-visual objects -- Part 10: Advanced Video Coding”, amendment containing definition the Multiview Stereo High profile for two-view video coding with support of interlaced coding tools and specifying an additional SEI message (the frame packing arrangement SEI message), March 9, 2010.

[ISO/IEC 14496-10:2014] ISO/IEC 14496-10:2014, “Information technology -- Coding of audio-visual objects -- Part 10: Advanced Video Coding”, amendment that specifies the coding of depth map data for 3D stereoscopic video, including a Multiview Depth High profile.), August 27, 2014

[KimLee2011] Young-il Kim, Yong Su Lee, et al., “The Performance Analysis of SVC image Service for Mobile IPTV System”, 13th International Conference on Advanced Communication Technology (ICACT), 2011, pages 1142-1145, February 13-16, 2011.

[Kramer2016] Richard A. Kramer, “An Introduction to the Problem: Interactive Free Viewpoint Live Multiview Video Streaming Using Network Coding”, Oregon State University, 2016.

[Morvan_deWithFarin2006] Yannick Morvan, Peter H. N. de With and Dirk Farin, “Platelet-based coding of depth maps for the transmission of multiview images”, Eindhoven University, 2006

[MüllerSchwarz2013] Karsten Müller, Heiko Schwarz, et al., “3D High-Efficiency Video Coding for Multi-View Video and Depth Data”, IEEE Transactions on Image Processing, Vol. 22, No. 9, pages 3366-3378, September 2013.

[OhmSullivan2005] Jens-Rainer Ohm, Gary Sullivan, “MPEG-4 Advanced Video Coding”, MPEG doc#: N7314, July 2005

NguyenNguyenCheung2010] Kien Nguyen, Thinh Nguyen, and Sen-Ching Cheung, "Video Streaming with Network Coding", Journal of Signal Processing Systems Archive Volume 59 Issue 3 at pages 319-333, June 2010.

[Ohm2009] Jens-Rainer Ohm, "MPEG Developments in Multi-view Video Coding and 3D Video", *Multiview & 3D Video Coding – EBU Workshop*, April 2009.

[Polycom2010] Polycom Whitepaper, "More Scale at Lower Cost with Scalable Video Coding", November 2010

[Rimac-DjljeNemčićVranješ2008] Snježana Rimac-Drlje1, Ognjen Nemčić, Mario Vranješ, "Scalable Video Coding Extension of the H.264/AVC Standard", 50th International Symposium ELMAR-2008, pgs. 9-12, September 10-12, 2008.

[SchierlNarasimhan2011] Thomas Schierl, Sam Narasimhan, "Transport and Storage Systems for 3-D Video Using MPEG-2 Systems, RTP, and ISO File Format", Proceedings of the IEEE, Vol. 99, No 4, pages 671-683, April, 2011.

[Smolic2008] Aljoscha Smolic, "Advanced Video Coding" subsection "Multiview Video Coding", MPEG-4, The Motion Picture Experts Group, MPEG doc #N9580, January 2008, June 16, 2016 article was accessed at: http://mpeg.chiariglione.org/standards/mpeg-4/advanced-video-coding.

[UittoVehkapera2013] Mikko Uitto, Janne Vehkapera, "Enhanced Quality Adaptation Strategies for Scalable Video", Signal Processing and Information Technology(ISSPIT), 2013 IEEE International Symposium, pages 74-79, 2013

[VetroWiegandSullivan2011] Anthony Vetro, Thomas Wiegand and Gary Sullivan, "Overview of the Stereo and Multiview Video Coding Extensions of the H.264/MPEG-4 AVC Standard", *Proceedings of the IEEE*, Vol. 99, No. 4, at pages 626-642, 2011.

[VideoBandwidthEstimates] Video bandwidth estimates, at link: http://mathscinotes.com/2012/05/high-definition-television-bandwidth-and-compression-math/, last visited Nov. 13, 2016

[WiegandSullivan2003] Thomas Wiegand, Gary J. Sullivan, "Overview of the H.264/AVC Video Coding Standard" IEEE Transactions on Circuits and Systems for Video Technology, Vol. 13, No. 7, July 2003

Appendix A – Glossary of Terms

3D MVV: 3-Dimensional Multiview Video also known as “2D plus delta” or “stereo” Multiview Video)

ADEM: Adaptation Decision Engine Module

AVC: Advanced Video Coding

B-Frame: Bidirectional Frame

CGS: Course-Grain quality Scalable coding

DIOMEDES: DIstribution Of Multi-view Entertainment using content aware DElivery Systems

DTH: Direct To Home

DVB: Direct Video Broadcast

DVB-CMS: Direct Video Broadcast – Content Management Server

DVB-H: Direct Video Broadcast-Handheld

DVB-T: Direct Video Broadcast-Terrestrial

FTV: Free viewpoint multiview TV

GOP: Group of Pictures

HDTV: High Definition TV

HEVC: High Efficiency Video Coding

IDR: Instantaneous Decoding Refresh

I-Frame: Intra Frame

IPTV: Internet Protocol Television

MGS: Medium-Grain quality Scalable coding

MPEG: Motion Picture Experts Group

MVC: Multiview-Video Coding

MVV: Multi-View-Video (also commonly shown as “Multiview Video”)

NAL: Network Abstraction Layer

NUT: Network abstraction layer Unit Type

OLFVmv: Optimized Live Free Viewpoint multiview video

PCR: Program Clock Reference

P-Frame: Predicative (and/or Prediction) Frame

P2P: Peer-to-Peer

P2P-MS: Peer-to-Peer Management Server

SPS: Sequence Parameter Set

SVC: Scalable Video Coding

VCL: Video Coding Layer

Appendix B – About the Author

Richard A. Kramer (Member IEEE) is a proven results-oriented R&D (Research and Development) leader with 30+ years of successes within the high-volume electronics and software industries. During his career, Richard has consistently translated customer opportunities, innovation and cross-functional execution into successful products, software, and technology within Fortune 100 and emerging high-tech start-up companies alike.

Richard is the President of SIS Development, Inc., a technical services company that provides expertise to product and software companies like Cisco, Google, Apple Computer, General Electric, United Technologies, Schneider Electric, Honeywell, FLIR Systems, and the like.

Prior to founding SIS Development, Inc., Richard served as General Manager-Technology and Vice President, Engineering at General Electric's GE Security division, where he led a progressive 300+ person R&D organization accounting for +$500M per year in revenue. At GE, Richard managed 16 geographically dispersed organizations, providing advanced enterprise, commercial, residential and real-estate solutions for the video surveillance, intrusion systems, alarm monitoring, and key control markets. Among his major accomplishments, Richard transitioned the organization from being numerous acquired companies into a single, highly-leveraged R&D organization with a clear strategy.

Before joining GE, Richard held top-technology leadership executive positions within respected innovation pioneering companies like Scientific-Atlanta (which was acquired by Cisco), Schlumberger Industries and start-up innovation leader Ivex

Corporation (IP enabled video surveillance solutions). At Scientific-Atlanta, he built and led the organization that developed and launched the company's first internally-designed high-volume video set-top box. Richard then took over all domestic set-tops (+3M units/year total, +$400M/year in revenue). Thereafter, he was promoted to a senior leadership position in a newly formed strategic planning organization, where he led a cross-functional organization dedicated to developing, launching and manufacturing the next generation "Advanced Video Systems". Richard started leading formidable organizations in 1990 after receiving a consistent series of promotions at Schlumberger Industries.

Richard holds a BSEE degree from the University of Toledo where he graduated Magna Cum Laude. He completed numerous Executive MBA courses at Emory University's top-10 ranked business school program, is a Six-Sigma Black Belt, is a member of IEEE, and he holds two patents.

Appendix C – Matlab Simulation Code and Results