

Czech Technical University in Prague
Faculty of Information Technology
Department of Digital Design



**Low-Latency Video Transmissions for Real-Time Collaboration
with a Scalable Hardware Acceleration**

by

Petr Žejdl

A thesis submitted to
the Faculty of Information Technology, Czech Technical University in Prague,
in partial fulfilment of the requirements for the degree of Doctor.

PhD programme: Informatics

Prague, August 2013

Thesis Supervisor:

Dr. Sven Ubik
CESNET – Czech Academic Network Operator
Žikova 4
160 00 Prague 6
Czech Republic

Copyright © 2013 Petr Žejdl

Abstract

Video transmissions are an expected driver application area of the future Internet. High resolution low-latency video transmissions allow a new exchange of information and real-time collaboration which was not available before due to unavailability of suitable hardware and capacity in computer networks. This kind of transmissions also sets new requirements on the underlying hardware as the number of bits transferred per second increases with the size of the transferred image but the network jitter and the processing latency have to be kept as small as possible.

This doctoral thesis describes work performed by the author which addresses the area of high-resolution low-latency video transmissions for real-time collaboration. In particular, a new and now patented technique based on asynchronous clock recovery for the receiver synchronization is presented. The technique allows to conduct very short latency transmissions with minimal hardware requirements over asynchronous packet based networks such as Internet. This makes the technique suitable in the transmissions for remote collaboration, where the low latency is extremely important.

The synchronization technique was experimentally verified by designing a proof-of-concept prototype platform for very low-latency high-resolution video transmissions called MVTP-4K (Modular Video Transfer Platform). Such platform is required for an interactive long-distance remote collaboration. The platform video processing latency is less than 1 msec. This is much smaller than with previous devices and makes it perfect for low-latency transmissions.

The final part of this doctoral thesis focuses on an evaluation of the proposed technology in a collaborative working environment. We present real-world use cases of remote real-time collaborations where the research results described in this doctoral thesis made an important contribution to the collaborative working environment. In particular, they have a verified impact on the productivity and brought a new style of working. The presented use cases involve enhancements in several applications in the film industry, e-Learning in medicine, art and culture. Several use cases were conducted over a distance of more than 10000 km, across continents. The observations along with practical experience are also presented.

Keywords:

High-definition Video, Low Latency, Remote Collaboration, Network Communication, Clock Synchronisation, FPGA.

As a collaborator of Petr Žejdl and a co-author of his papers, I agree with Petr Žejdl's authorship of the research results as stated in this dissertation thesis.

.....
Jiří Halák

.....
Jiří Navrátil

Acknowledgements

First of all, I would like to express my gratitude to my dissertation thesis supervisor, Dr. Sven Ubik. He was a constant source of encouragement and insight during my research and helped me with numerous problems and professional advancements.

Furthermore I would like to thank to all of my collages and co-authors for their excellent cooperation and support, especially to Jiří Halák whose work enabled my research. He spent many hours with me during the building of prototypes, simulations, testing and in-field evaluations. Also I would like to Dr. Jiří Navrátil who spent together with my supervisor many hours with planning of the long-distance experiments and for many comments and recommendations I received from him. And I would like to thank to Dr. Jan Gruntorád (director of CESNET association) who provided convenient and flexible environment for my research activities.

This thesis would never have been possible without the support of my wife Petra and my sons Jan and David. My greatest thanks go to them for their infinite patience and care!

Many people and organizations provided generous support in the form of equipment loans, technical cooperation and recommendations as follows: Cinepost company at Barrandov Studios in Prague, Czech Republic; Visual Unity Ltd. in Prague, Czech Republic; Universal Production Partners (UPP) company in Prague, Czech Republic; Masaryk hospital in Usti nad Labem, Czech Republic; KEK, the High Energy Accelerator Research Organization in Tsukuba, Japan and University of California, San Diego. Also I would like to thank to all CineGrid members and CineGrid networks and all the others I forgot to mention. . .

This work was supported by the Ministry of Education, Youth and Sports of the Czech Republic under the research intent MSM6383917201 “Optical Network of National Research and Its New Applications”.

This work was supported also by the Technology Agency of the Czech Republic within the POVROS project (TA01010324) and partially supported by the following international projects funded by the European Commission: LOBSTER, GÉANT GN2 and GÉANT GN3.

Contents

Abstract	iii
List of Figures	xiii
List of Tables	xv
1 Introduction	3
1.1 Motivation	3
1.1.1 Remote collaboration in Multimedia Industry	4
1.1.2 Remote collaboration and enhancements in e-Learning in Medicine	4
1.1.3 Remote collaboration in Art, Culture, Science and Engineering . . .	5
1.2 Problem Statement	6
1.2.1 Real-time processing of high-data volume	6
1.2.2 Reducing processing latency	7
1.2.3 Receiver to sender synchronization	7
1.3 Contributions of the Thesis	8
1.4 Structure of the Thesis	9
2 Background and State-of-the-Art	11
2.1 Theoretical Background	11
2.1.1 Video Transmissions	11
2.1.2 Serial Digital Interface (SDI)	12
2.1.3 Video Data Encapsulation and Packetisation	15
2.1.4 Packet Loss and Network Jitter	16
2.1.4.1 Packet Loss and Video Frame Loss	16
2.1.4.2 Network Jitter	18
2.1.5 Playout Problem	19
2.1.6 Clock Recovery	21
2.1.6.1 Synchronous Clock Recovery	22
2.1.6.2 Asynchronous Clock Recovery	22
2.1.6.3 Notes on video Synchronization	24
2.2 Related Work	25
2.2.1 NTT Network Innovation Laboratories	25

2.2.2	UltraGrid	26
2.2.3	LOLA – LOw LAtency audio visual streaming system	26
2.2.4	Commercial devices	27
2.2.5	Summary	27
2.3	My Previous Results	29
2.3.1	Background	29
2.3.2	Modular Traffic Processing Platform (MTPP10)	29
2.3.3	Modular Traffic Processing Platform 40 Gb/s (MTPP40)	31
2.3.4	Summary	32
3	Contributions	33
3.1	A technique for receiver synchronization in video streaming with short latency over asynchronous networks	33
3.1.1	Introduction	33
3.1.2	Architecture Overview	35
3.1.3	Summary	37
3.2	Proof-of-concept prototype for low latency video high-resolution transmission	39
3.2.1	Introduction	39
3.2.2	Architecture Overview	40
3.2.2.1	Firmware architecture	40
3.2.2.2	Plug-in module architecture	42
3.2.2.3	A SystemACE based architecture simplifying partial reconfiguration	43
3.2.2.4	10 Gb/s network interface	44
3.2.2.5	Unified software framework for packet classification and filtering configuration	45
3.2.3	Evaluation	47
3.2.3.1	4K transmission in a loop over 14602 kilometers from Prague to Chicago	47
3.2.3.2	HD transmission in loop over 35200 kilometers from Prague to Japan	49
3.2.4	Summary	51
3.3	Prototype evaluation in a collaborative working environment	53
3.3.1	Remote collaboration in multimedia industry	53
3.3.2	Remote collaboration and enhancements in e-Learning in medicine	55
3.3.3	Remote collaboration in art, culture, science and engineering	58
4	Conclusions	61
4.1	Summary	61
4.2	Future Work	62

5 Publications Included in Thesis	65
5.1 J. Halak, S. Ubik, and P. Zejdl. Receiver synchronization in video streaming with short latency over asynchronous networks	65
5.2 J. Halak, M. Krsek, S. Ubik, P. Zejdl, and F. Nevrela. Real-time long-distance transfer of uncompressed 4K video for remote collaboration	69
5.3 J. Halak, S. Ubik, and P. Zejdl. Scalable embedded architecture for high-speed video transmissions and processing	77
5.4 S. Ubik, J. Navratil, P. Zejdl and J. Halak. Real-Time Stereoscopic Streaming of Medical Surgeries for Collaborative eLearning	84
5.5 P. Zejdl, S. Ubik, V. Macek, and A. Oslebo. Traffic classification for portable applications with hardware support	90
5.6 J. Halak, S. Ubik, and P. Zejdl. A DEVICE FOR RECEIVING OF HIGH-DEFINITION VIDEO SIGNAL WITH LOW-LATENCY TRANSMISSION OVER AN ASYNCHRONOUS PACKET NETWORK, International Patent	100
Bibliography	107
Publications of the Author	113
List of Glossed Papers	113
Relevant Refereed Publications	114
Remaining Refereed Publications	115
Unrefereed Publications	115
Patents and Utility Patents	116
Prototypes	117
A Prototypes	119
A.1 MTPP-10	119
A.2 MTPP-40	120
A.3 MVTP-4K	121
B List of abbreviations	123

List of Figures

2.1	Network Video Streaming Dataflow	12
2.2	A 4K image displayed on a 4K LCD Monitor (Astro Design DM-3400) in the CESNET labratory. The four 2K HD-SDI quadrants can be clearly seen.	13
2.3	2K HD-SDI Line Format. The numbers on the vertical axis correspond to lines, the numbers on the horizontal axis correspond to samples within each line. EAV (End of Active Video) and SAV (Start of Active Video) are timing references.	14
2.4	MPEG4 video stream at constant bit-rate 200kbps [1]. The spikes in target bit rate correspond to key frames which contain a larger amount of data to be transmitted than the other frames.	18
2.5	The Playout Problem. The black horizontal bars indicate the arrival times of packets at different locations between sender and the playout device. The dashed lines correspond to data packets. The red cross illustrates a packet which was lost in the network. There are two packets which arrived out of order and they have to be swapped. The last (bottommost) packet suffers from excessive delay in the network and arrives later than the expected playout time minus the buffer delay meaning that it can not be display in time and must therefore dropped at the receiver.	20
2.6	Clock recovery block diagram [2]	22
2.7	Adaptive clock recovery	23
2.8	Modular Traffic Processing Platform Architecture	30
2.9	Hardware Architecture of the 40 Gb/s Modular Traffic Processing Platform	31
3.1	Synchronization block diagram	35
3.2	An example of receiver synchronization without frame alignment demonstrated with Sony SRX 4K projector. (Note the white line in the middle of the screen.)	36
3.3	Hardware architecture	40
3.4	Firmware architecture of the prototype implementation	41
3.5	List of the currently loaded plug-in modules for video processing. The list is provided by <code>mtp</code> tool executed in the embedded Linux system.	42
3.6	FPGA Configuration using System ACE controller	43
3.7	Ethernet Interface Block Diagram	45

3.8	Architecture of transparent and extensible packet classification	46
3.9	Remote loopback test (Prague-Chicago-Prague loop, 14602 kilometers) . .	47
3.10	The processing delay measured in Prague-Chicago-Prague loop. Negative values mean that the lines arrived before they should be sent to the output.	48
3.11	Regulation system response in Prague-Chicago-Prague loop. During the first 10 seconds the system is adapting to the sender's clock frequency. . . .	48
3.12	Remote loopback test (Prague-KEK in Japan-Prague loop, 35200 kilometers)	50
3.13	Inter-packet arrival (delta) times. The left plot shows the raw delta time measurement, the right plot shows the sorted values.	51
3.14	Schematic diagram of the network connection used during the remote color grading demonstration	53
3.15	The screen configuration in lecture hall during the remote stereography demonstration	54
3.16	The da Vinci Surgical System, from left to right: The operator console, the operating robot, a detailed view of the robot arms with the surgical instruments (third instrument from the left is the stereoscopic camera). . .	56
3.17	Remote audience viewing and discussing the surgery during the 5th International Congress of Mini-invasive and Robotic Surgery in Brno (Czech Republic).	57
3.18	A view of the remote audience room during the CineGrid 2011 presentation. The image looks blurred because the images for both eyes appear superimposed.	58
A.1	10 Gb/s Modular Traffic Processing Platform (MTPP-10) prototype	119
A.2	40 Gb/s Modular Traffic Processing Platform (MTPP-40) prototype	120
A.3	MTPP-40 Optical Loopback Test Setup	120
A.4	The first Modular Video Transfer Platform (MVTP-4K) prototype	121
A.5	MVTP-4K Optical Loopback Test Setup with 4K LCD Monitor (Astro Design DM-3400)	121

List of Tables

2.1	HD-SDI formats and frame sizes. The <i>Frame</i> shows the size of a visible part of the frame. The <i>Full Frame</i> shows the size including the non-visible blanking parts. The <i>Frame data rate</i> is a bandwidth required for transferring the visible part of the frames. The two streams marked by * require dual-link HD-SDI or 3G interface.	15
-----	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----

Chapter 1

Introduction

Video transmissions are an expected driver application area of the future Internet. The bandwidth associated with core networking was observed, on average, to be doubling every eighteen months. If the current trend continues, then this translates to an increase in traffic of a factor of 10 by 2015 compared to 2010 and by a factor of 100 by 2020 [3]. Internet video is expected to increase from 40 % to approximately 61 % of all consumer Internet traffic in 2015 and will therefore be the dominant type of communications over the internet. The current generation of fiber optics can operate at speeds over 40 Gb/s, and 100 Gb/s Ethernet equipment is already commercially available [4] which allows to transmit hundreds high-speed streams or high resolution video channels simultaneously using Internet or dedicated fibres today (year 2013).

Image resolution is also increasing over time. Better-than-high-definition-resolution video, such as 4K (approximately 4 times the resolution of HDTVs) or even higher, is already used in some areas such as film industry and scientific visualization. In 2012, Japan Broadcasting Corporation (NHK) used the London Olympic Games as a test platform for Super Hi-Vision transmission – a new technology that uses 8K resolution cameras, 8K projectors and a 22.2 channels audio system. The Super Hi-Vision is approximately 16 times the resolution of HDTVs. This resolution is close to the detail level of 15/70mm IMAX film which has a resolution similar to 10K [5]. Currently, there are organizations conducting research on holograms and there is a demand for 16K or even higher resolutions (approximately 64 times the resolution of HDTVs).

Presentation of ultra high resolution video is now well possible using rendering devices with the corresponding resolution (4K projector, Super Hi-Vision projector), using tiled displays (e.g., SAGE [6, 7]) or multi-dimensional systems (CAVE [8, 9]).

1.1 Motivation

If we achieve a higher resolution and lower latency than in the current video transmission systems, we enable a new exchange of information and collaboration which was not available before or which was difficult to arrange. A very important area of applications is remote real-time collaboration where the productivity of a distributed team can be significantly

increased when the high resolution video signal is transferred over the network in real time with minimum latency and high resolution.

The following lines present real-world use cases of remote real-time collaborations where the technology allowing low-latency video transmissions has a verified impact on the productivity or brought a new style of working. All the use cases presented here were evaluated and are described along with observations and practical experience in Section 3.3.

The use cases include:

- Remote real-time collaboration in Multimedia Industry
- Remote real-time collaboration and enhancements in e-Learning in Medicine
- Remote real-time collaboration in Art, Culture, Science and Engineering

The research of such technology allowing a low-latency video transmission is described in this doctoral thesis. The research led to the development of a new and now patented device called MVTP-4K (Modular Video Transfer Platform) [10, 11] which was used in the all described field trials. The results are discussed in detail in Section 3.3.

1.1.1 Remote collaboration in Multimedia Industry

The technology for the low-latency and high-quality transmissions of image and sound can enable effective remote real-time collaboration specially in post-production of movies where tasks such as color grading, stereography or image restoration are performed. The post-production often involves several key parties who need to be able to discuss and control the process - the director, the producer, the editor, and several technical experts - the colorist, the stereographer, the sound master, etc. However, the participants are often very busy working on multiple projects in parallel in different locations distributed across the world. Therefore, it is difficult for them to meet in one place for a collaborative session.

The low-latency and high-quality transmissions for post-production allow to connect the key parties together without the need for traveling. As the capacity of optical networks is increasing, uncompressed transmissions of original content with minimal latency is possible and preferred. The team can work and see the image across 10000 km in real-time without any observable image degradation.

This would save a lot of traveling time, speed up the whole process and therefore save money and it would also be environmentally more friendly. Highly secure encryption would be a required feature, as it is now a standard for content distribution to cinemas on hard disks.

1.1.2 Remote collaboration and enhancements in e-Learning in Medicine

Minimally invasive surgery is a modern surgical technique where the surgery is performed through small incisions. This includes laparoscopic surgeries which makes use of images displayed on monitors to magnify the surgical elements. The latest advancement in minimally

invasive surgeries involves robotic surgery, or more precisely called robotically-assisted surgery, which utilizes a robotic system to aid in surgical procedures.

Robotic surgery brings several advantages to modern surgery techniques - precision, smaller incisions, decreased blood loss and consequently quicker healing time. Modern robotic surgery (e.g., the da Vinci surgical system [12]) makes use of stereoscopic video views of surgical elements for the better precision. The signals from this camera can be captured and used for E-health applications, such as remote medical students training or presentations of surgical procedures on symposia where are medical experts presenting more specialized interventions. For such E-health applications, we need to transfer the stereoscopic high-resolution signal from the surgery robot to the audience over potentially large distances at high quality and with short latency. The latter is particularly important to provide interactive experience in communication between the audience and the doctors.

1.1.3 Remote collaboration in Art, Culture, Science and Engineering

Stereoscopic (3D) models and visualizations are used in many fields of research, in engineering design and even in art and humanities. Such use-cases involve:

- Remote visualization of 3D museum artefacts
- Remote access to scientific 3D models
- Real-time streaming to large visualization devices (SAGE, CAVE)

Rendering complex 3D models is computationally expensive. If we can transfer the two dimensional projection of 3D models in real-time across a computer network in high quality with low latency and allow people to interact with the model, it is not necessary to build a graphical rendering computer cluster in each place where the 3D model shall be displayed.

Another purpose of remote 3D visualizations is to gain access to models which cannot be moved physically. Real-world models may not be movable due to their size or weight and computer models may have intellectual property rights applied restricting their sharing in data form (while visualization of the model can be shared).

Probably the most immersive systems for visualization and presentation is the multi-dimensional system CAVE (Cave Automatic Virtual Environment) or tiled displays SAGE (Scalable Adaptive Graphics Environment). CAVE presents a 3D image on screens completely surrounding a small group of people. Movements of one person can be tracked by sensors and the image is adjusted accordingly to give the viewer the impression of a virtual reality. SAGE is a flat high-resolution video wall composed of a group of LCD panels. It is often used by researchers to present multiple visualizations together on a large working space. Some devices can present 3D images.

Different teams collaborating on the same project may have different visualization environments. An important aspect is the processing the processing and transmission of large volumes of data with low latency.

1.2 Problem Statement

Following the discussion in the previous section, there are several problems that need to be resolved in order to enable the targeted applications:

- Real-time processing of high-data volume
- Reducing processing latency
- Receiver to sender synchronization

The following sections briefly discuss these problems. For a detailed explanation, please, refer to Chapter 2 on page 11.

1.2.1 Real-time processing of high-data volume

For the ultimate quality, required for instance in the color-grading process in film post-production, working with a signal that has not been compressed is preferable. The data volume ranges from approximately 1 Gb/s for one 2K stream at 24 fps (4:2:2 chroma sub-sampling [13] such as used in many high-end digital video formats) to 10 Gb/s for a 4K stream at 60 fps. Higher data volumes are required for transmitting 3D or high color depth streams. Overhead in packet headers needs to be added.

The capacity of current optical networks allows to transfer such bandwidths. We have been working in a community of partners interconnected by a 10 Gb/s high-speed networking environment called GLIF (Global Lambda Interchange Facility) [14]. It is an international virtual organization that promotes the paradigm of lambda networking to support distributed data-intensive applications. We believe that high bandwidth availability will be a part of future networking advanced by gradual shifting of network functionality from the electrical to the optical domain.

Real-time processing of multi-gigabit data rates is difficult on PC-based platforms with standard operating systems not designed for real-time operation. We are looking for a real-time design that is scalable to higher data rates (such as 8K), higher network speeds (such as 40 and 100 Gb/s), which can process image data with the lowest added latency (a line rather than a frame based processing) with minimal data buffering and that can be integrated with commonly requested video processing functions, such as encryption, transcoding or compression. This implies highly parallel and truly real-time data paths with minimal added latency. The DSP (Digital Signal Processor) and FPGA (Field Programmable Gate Arrays), often combined together, are the standard technologies in this area.

Another requirement is encryption of the transmitted data. The copyright holders of the material used in multimedia industry often require highly secure encryption to be used for the transmission. Encryption can easily be implemented by the addition of a cryptographic plug-in module in the target design. This is possible by modular design of our prototype platform and is described in detail in Section 3.2.2.2.

1.2.2 Reducing processing latency

The end-to-end (one-way) latency of the video transmission consists of the video stream processing delay and network delay caused by signal propagation through the electrical and/or optical cables and network interconnecting devices.

For an interactive feeling of remote collaboration, the end-to-end latency between all locations should be low and close to each other. The empirical experience has shown that the limit for the user to feel that the system is reacting instantaneously is between 100 and 200 ms [15]. Below this limit, the users will experience transparent interactivity. Above this limit, they would feel a delay and will try accommodating this delay into their thinking with negative effects on productivity. The ITU recommended limit is 150 ms [16].

The inevitable network propagation delay is approximately 50 ms across Europe or the US and 100 to 120 ms from Europe to Japan or to the West Coast US. In remote visualization, the delay between a movement of a control device and the response in visualization is the round-trip network delay which is typically twice the one way network delay.

Therefore, it is important to keep any additional delay caused by processing as low as possible. The processing delay depends on the type of encoder and decoder used to manipulate the image and on the receiver rendering (play-out) technology. The receiver play-out technology aims to smooth the delay variations (jitter) between packet arrivals for a given stream. These techniques commonly involves the stream playout (de-jitter) buffers. However, these buffers introduce an additional delay in order to produce a constant video playout rate. Buffering of one frame at 24 frames per second (common format used in multimedia industry) adds another 41 milliseconds of delay and should be avoided if possible.

1.2.3 Receiver to sender synchronization

Real-time video streaming requires that the speed of video rendering on the receiver side matches the speed of source video on the (network) sender side. However, when the sender and receiver are connected over an asynchronous network (such as Ethernet), the receiver cannot directly synchronize its video clock with the sender video clock. The clock has to be derived from the incoming network stream containing some periodic information.

In practice, the bit rate of video streams (especially when using compression) varies with time. Therefore, it is difficult to distinguish if the variation in the periodicity of the incoming network stream is caused by the video encoding/compression or by the network jitter.

We are looking for a receiver synchronization technique that is scalable to higher resolutions, requires the smallest possible amount of buffer memory and keeps the processing delay to a minimum, is resilient to the network jitter and still allows smooth image playback. This allows direct implementation in an FPGA without the need for the external memories.

1.3 Contributions of the Thesis

The contributions of this doctoral thesis are discussed in detail in Chapter 3. Here, we briefly present the main results:

- **A technique for receiver synchronization in video streaming with short latency over asynchronous networks**

I proposed a new and now patented technique based on an asynchronous clock recovery method for the receiver to sender video synchronization. This technique makes use of small memory buffers (much smaller than the buffer required for one image frame). Thus, this allows a direct FPGA implementation and is adding a minimal processing delay less than 1 msec. This is much smaller than with previous devices. Due to the small processing delay, this technique allows to conduct transmissions with very low latency with minimal hardware requirements over asynchronous networks such as Ethernet and Internet.

- **Proof-of-concept prototype for low latency video high-resolution transmission**

The second contributions describe the proof-of-concept architecture of a prototype developed during the research. The proposed synchronization technique is a key part of the prototype. The prototype is capable of low-latency video transmissions with resolutions up to 4K in stereoscopic mode (3D). The following results were achieved with this prototype implementation:

- The video processing is added as series of reconfigurable modules with the common interface. I proposed an architecture simplifying the module preparation and reducing the complexity of the hardware required for the dynamic partial reconfiguration.
- I proposed a new extendable software framework for packet filtering and classification configuration. The framework is able to configure different hardware or different network monitoring cards with classification rules specified in only one language.

- **Prototype evaluation in a collaborative working environment**

The last but not least contribution of this doctoral thesis is an evaluation of the proposed technology in a collaborative working environment. We conducted several real-world demonstrations of real-time remote collaborations with focus on applications in the film industry, e-Learning in medicine, art and culture, where the contributions described in this doctoral thesis have a verified impact on the productivity and brought a new style of working. Several demonstrations were conducted over a distance of more than 10000 km, across continents. The observations along with practical experience are also presented.

1.4 Structure of the Thesis

The thesis is organized into 5 chapters as follows:

1. *Introduction*: Describes the motivation behind our efforts together with our goals. The brief summary of the previous results and contributions of this doctoral thesis is also presented here.
2. *Background and State-of-the-Art*: Introduces the reader to the necessary theoretical background, surveys the current state-of-the-art technologies, summarizes the notable previous results and briefly describe my previous results.
3. *Contributions*: Summarizes the contributions of this dissertation thesis.
4. *Conclusions*: Summarizes the results of our research, suggests possible topics for further research, and concludes the thesis.
5. *Publications Included in the Thesis*: This section contains selected publications which address the research described in this doctoral thesis.

Chapter 2

Background and State-of-the-Art

2.1 Theoretical Background

This section provides an overview and the basic background and concepts in the field of low-latency video transmissions.

2.1.1 Video Transmissions

As we discussed in the previous chapter, transmitting video across a network requires to deal with several issues such as network jitter, latency and receiver synchronization. The whole transmission problem can be seen as a conversion from a synchronous video stream at constant bit rate to a variable bit rate stream after encoding, transmitting over an asynchronous network (packet based), decoding and converting back to a synchronous video stream at the receiver. The receiver has to adapt its speed to the speed of the sender (rate synchronization with clock recovery) in order to provide the same synchronous and uninterrupted video stream as was captured by the sender.

A schematic representation of such conversion is shown in Figure 2.1. We consider the sender as a device capturing digital video signal (described in the the following section), converting and sending the data to the network and the receiver as a device receiving data from the network and sending video signal to an output. The video signal is captured by the sender and processed by encoder, where blanking intervals are removed and raw video data are optionally compressed. The encoded video stream is encapsulated into a suitable network transport protocol, split into packets and send over the network. In the receiver the apposite process follows. The video stream is reconstructed from the received packets. The playout buffer is used to remove network jitter and reconstruct the original synchronous video stream. The video stream is decoded into raw video data and is displayed.

It is important to provide a synchronous video stream even if some of the data were lost (e.g. due to network congestion). If there is a glitch in the video stream, the rendering device will lose synchronization and it will take up to several seconds to resynchronize itself without rendering any usable data. This behavior is unacceptable in low-latency transmission for remote collaboration and has to be avoided.

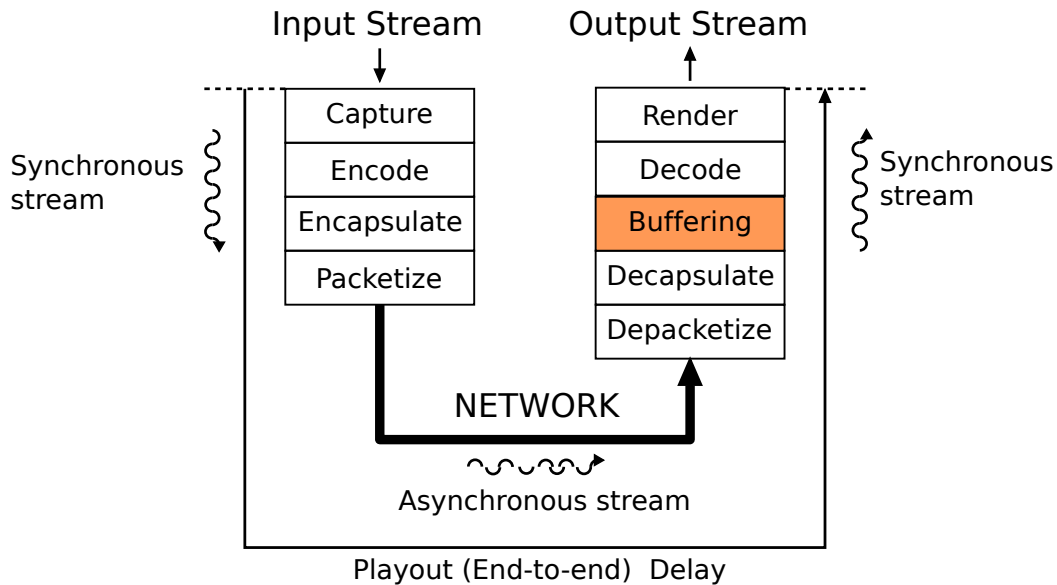


Figure 2.1: Network Video Streaming Dataflow

In the following text we understand the network as a network based on asynchronous Ethernet technology. Asynchronous Ethernet is currently more frequently deployed in 10 Gb/s networks than synchronous SONET/SDH, due to its simplicity and therefore lower cost. Ethernet will likely play an even more important role in future 40 Gb/s and 100 Gb/s consumer networks, although often encapsulated in synchronous Optical Transport Networks (OTN) [17] using a DWDM (dense wavelength-division multiplexing) optical transmission technology [18].

2.1.2 Serial Digital Interface (SDI)

The professional multimedia industry including TV broadcast studios is using coaxial cables to distribute video signals. The first transferred signals were analog, later by the end of 20th century, the first digital devices appeared. To keep the existing coaxial cable installations, engineers developed a method for transferring digital content across the legacy analog studio networks. The method involves a new interface – the Serial Digital Interface (SDI) and is described in the series of standards maintained by the Society of Motion Picture and Television Engineers (SMPTE) [19].

The SDI interface involves several related interfaces which are specified in the relevant SMPTE documents. Commonly used SDI interfaces are:

- **SD-SDI** [20]: Used to transport uncompressed standard-definition digital video with resolutions 480i and 576i. The link speeds are 143/270/360 Mb/s.
- **HD-SDI** [21, 22]: Used to transport uncompressed high-definition digital video with resolutions 720p, 1080i and 2K. The link speed is 1.485 Gb/s.

- **Dual Link HD-SDI [23] or 3G-SDI [24]:** Used to transport uncompressed high-definition digital video with resolutions 1080p and 2K. The link speed is 2.970 Gb/s.
- **6G-SDI [25]:** The SMPTE standardization process is pending. However, the devices using this standard are already deployed, transporting digit video with resolution up to 4K. It is backward compatible with the previous standards, the link speed is 5.940 Gb/s.

The 'i' and 'p' denominator at the end of the resolution size states whether the video format is interleaved or progressive (non-interleaved). The SDI format is using $Y'C'_bC'_r$ colorspace with 4:2:2 or 4:4:4 subsampling or R'G'B' colorspace. The color width is 10 or 12 bits. The defined frame rates are 24, 25, 30, 50 and 60 frames/sec for compatibility with PAL systems and its fractional counterparts (multiplied by 1/1.001 constant) for compatibility with NTSC systems.

In the following text we will focus on the high-resolution digital video connected through the HD-SDI interface. This also includes the dual-link HD-SDI and 3G-SDI. The dual-link HD-SDI is nothing else than the two bundled HD-SDI connections allowing higher frame rates. The 3G-SDI is almost the same like dual-link HD-SDI but using only one physical connection at double link-rate.



Figure 2.2: A 4K image displayed on a 4K LCD Monitor (Astro Design DM-3400) in the CESNET laboratory. The four 2K HD-SDI quadrants can be clearly seen.

Ultra high-resolutions such as 4K resolution are typically transferred in four quadrants, each in 2K format carried over a separate dual-link HD-SDI or 3G interface. The 4K

layout is illustrated in Figure 2.2. A 2K test image is electrically repeated four times and connected to the 4K monitor.

Figure 2.3 shows the physical line format of a 2K frame (one quadrant of the 4K frame) carried by the HD-SDI interface. Each frame consists of 1125 lines, which includes 1080 active lines containing visible data. Each line consists of 2750 samples, which includes 2048 active samples carrying 20bit color samples. Here, we allow one simplification: For uncompressed 4:2:2 signals, the HD-SDI interface consists of two parallel bit streams one stream is defined as the Luma (Y) data channel, and the second stream is the Color-Difference ($C'_b C'_r$) data channel. These data channels are multiplexed to form the serial data stream. The two data channels must be perfectly synchronized and no skew is allowed between them.

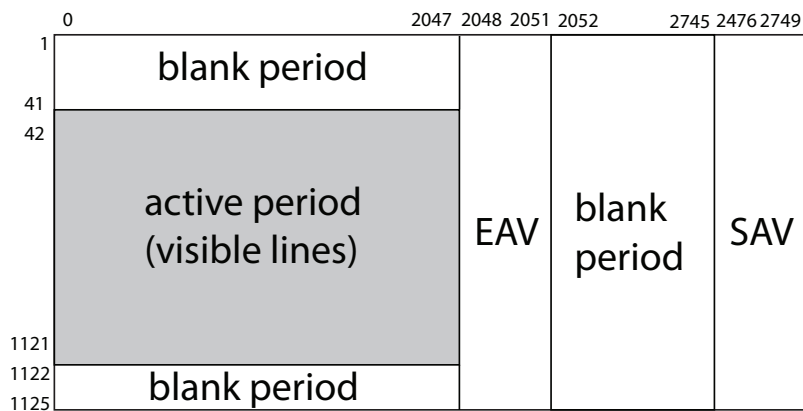


Figure 2.3: 2K HD-SDI Line Format. The numbers on the vertical axis correspond to lines, the numbers on the horizontal axis correspond to samples within each line. EAV (End of Active Video) and SAV (Start of Active Video) are timing references.

The format presented in Figure 2.3 is used in digital cinema distribution systems, where the 2K picture resolution is the standard [22]. Other resolutions differ in the number of samples and active lines. The total number of lines is the same for all resolutions and is equal to 1125. The number of samples varies with the frame rate. The product of number of samples, number of lines and frame rate is called the interface sampling frequency and is equal to 74.25 MHz for all resolutions and frame rates in the HD-SDI standard. When a dual-link HD-SDI or 3G interface is used the interface sampling frequency is doubled to 148.5 MHz as the amount of data transferred per second is doubled.

Table 2.1 summarizes frame sizes and required bandwidth for HD-SDI 4:2:2 10bit $YC'_b C'_r$ color streams. The sample data rate is the bandwidth required for transmitting only the color samples. All the formats except 1920x1080 at 50 and 60 Hz can be transferred by a single-link HD-SDI interface. A dual-link HD-SDI or 3G interface allows to carry 1920x1080 resolution at 50 and 60 Hz because the interface data bandwidth is a factor two higher. The additional bandwidth in dual-link interfaces also allows to transfer 4:2:2 12bit $YC'_b C'_r$, 4:4:4 10 or 12bit $YC'_b C'_r$ or 4:4:4 RGB formats [23].

Image Format	Frame [MB]	Frame Rate Frames/sec	Samples x Lines	Full Frame size [MB]	Frame data rate [Gb/s]
1280 x 720	2.2	60	1650 x 750	2.9	1.11
		50	1980 x 750	3.5	0.92
		30	3300 x 750	5.9	0.55
		25	3960 x 750	7.1	0.46
		24	4125 x 750	7.4	0.44
1920 x 1080	4.9	60	2200 x 1125	5.9	2.49 (*)
		50	2640 x 1125	7.1	1.65 (*)
		30	2200 x 1125	5.9	1.24
		25	2640 x 1125	7.1	1.04
		24	2750 x 1125	7.4	1.00
2048 x 1080	5.3	24	2750 x 1125	7.4	1.06
4096 x 2160	21.1	24	5500 x 2250	29.5	4.25

Table 2.1: HD-SDI formats and frame sizes. The *Frame* shows the size of a visible part of the frame. The *Full Frame* shows the size including the non-visible blanking parts. The *Frame data rate* is a bandwidth required for transferring the visible part of the frames. The two streams marked by * require dual-link HD-SDI or 3G interface.

2.1.3 Video Data Encapsulation and Packetisation

The purpose of stream packetisation is to split a multimedia stream into data segments which are sent to an end-user through a network. Depending on the type of multimedia source, additional information has to be added to the segments for correct stream reassembly at the end-user side. Such additional information involves type of the stream (there can be several different streams transferred together such as video, audio and subtitle stream), stream format, resolution, playback speed and timing information (timestamps or sequence numbers). The latter is essential for reconstructing the stream video data in correct sequence and to guarantee smooth playback.

Then a network protocol is used to send the data segments through a network. The following set summarizes the typical requirements for multimedia delivery over a network:

- Low-latency delivery for real-time transmissions
- Reliability, in-order delivery, absence of losses if perfect stream quality is required
- Multicasting (optionally) for simple stream duplication, typically used in a big auditoriums equipped with several displaying devices presenting the same stream or for distribution over wide area such as commercial IPTV multicasting.

In packet based networks such as Ethernet or more generally Internet it is difficult to fulfill the all requirements at the same time. Depending on the nature of a multimedia

stream and quality requirements a tradeoff has to be found between low-latency and high reliability. The reliability is achieved by retransmitting the data which were corrupted or by embedding a forward error correction. For networks, where communication is unidirectional (e.g. Satellite TV), forward error correction is the only option.

Network protocols for multimedia transmissions can be divided into two basic groups based on the usage of the network layer: Datagram based and stream based.

Datagram based protocols

The most widely known example of such a protocol is the User Datagram Protocol (UDP), where the multimedia stream is sent in a series of small datagrams (packets). Packets are sent on a best effort basis, correct packet order and delivery is not guaranteed by the network. If some degree of reliability is required, it has to be established at the higher protocol level, e.g. by forward error correction (FEC).

UDP is rarely used alone, often a higher protocol is encapsulated into UDP datagrams. The Real-time Transport Protocol (RTP) [26] and the Real-time Transport Control Protocol (RTCP) are examples of protocols built on top of UDP and specifically designed to transport multimedia streams over networks.

UDP is typical protocol used in unidirectional video transmissions or video multicasting.

Stream based protocols

Such as the Transmission Control Protocol (TCP) and the Stream Control Transmission Protocol (SCTP) [27]. These protocols protocol guarantee reliable in-order delivery. However, the reliability is accomplished by waiting for a receiver's acknowledgment and retransmission is performed in case of timeout. Retransmissions may lead to significant increase in latency. While it is acceptable for an offline multimedia transmission, the quality of real-time transmissions where the low-latency is important may be decreased.

TCP can be directly used to transfer multimedia files or to transfer a higher protocol such as HTTP with a dynamic adaptive streaming [28], based on the available network bandwidth and the end-user requirements.

2.1.4 Packet Loss and Network Jitter

2.1.4.1 Packet Loss and Video Frame Loss

Due to the nature of video signals, network video streams exhibit a bursty character. In particular, when a real-time video signal is packetized, only the visible part of video lines are encoded and blanking areas are omitted. Such a stream exhibits bursts of packets (visible lines) followed by a pause (blanking area). Compressed streams also have similar characteristics where key-frames are often considerably larger than differential frames. Depending on the other network traffic and QoS (Quality of Service) settings, transitions from constant traffic rate to traffic bursts can interfere with other network streams (and the congestion control mechanisms of the other streams) and increase packet jitter and packet

loss. The following text gives brief description of packet loss, frame loss and network jitter and can be found in [1], where it is described in details.

Packet Loss

Packet loss is usually calculated on the basis of packet identifiers, such as the packet sequence number or the line number. The packet identifiers are also used to resolve the packet reordering problem. In the context of video transmission it is not only interesting to know how many packets were lost, but also which kind of data is in the packets. E.g., the MPEG-4 codec defines four different frame types and also some generic headers. For details see the MPEG-4 Standard [29]. Since it is very important for video transmissions to know which kind of data was lost (or not) it is necessary to distinguish between the different kind of packets. Evaluation of packet losses should be done by type (frame type, header). Packet loss measured by the quantity $PacketLoss_T$ defined in equation 2.1 is expressed in percent.

$$PacketLoss_T = \frac{n_{Trecv}}{n_{Tsent}} 100\%, \quad (2.1)$$

where:

- T : is the type of the packet
- n_{Trecv} : number of T packets sent packets
- n_{Tsent} : number of T packers received

Video Frame Loss

Key-frames used in compressed streams are typically considerable larger than differential frames and are causing spikes (packet bursts) when transmitted over a network. This is true not only in the case of variable bit rate streams, but also in the case when a constant bit rate stream is requested, since the term constant applies to a short time window. Such situation is depicted in Figure 2.4, where the traffic bursts related to the transmission of key-frames are clearly seen. In principle the frame loss rate can be derived from the packet loss rate. But this process depends on the capabilities of the actual video decoder in use. Some decoders can process a frame even if some parts are missing. Furthermore, whether a frame can be decoded depends on which type of the packets were lost. If the first packet is missing, the frame can almost never be decoded. Thus, the capability of a given decoder has to be taken into account in order to calculate the frame loss rate. It is calculated separately for each frame type.

Frame loss is defined similarly to packet loss in and is expressed in percent:

$$FrameLoss_T = \frac{n_{Trecv}}{n_{Tsent}} 100\%, \quad (2.2)$$

where:

- T : is the type of the frame (I, P, B, S)

$n_{T_{recv}}$: number of T frames sent

$n_{T_{sent}}$: number of T frames received

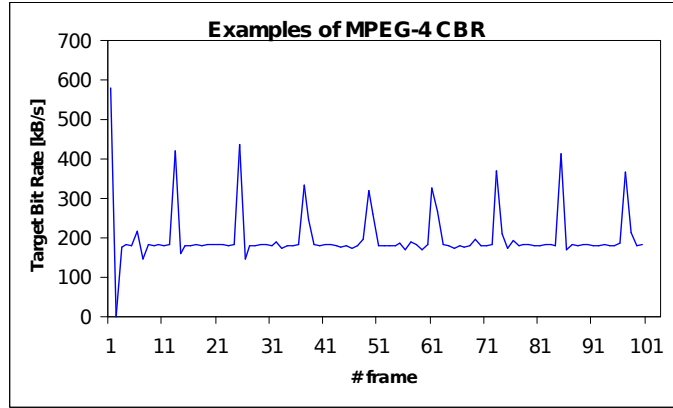


Figure 2.4: MPEG4 video stream at constant bit-rate 200kbps [1]. The spikes in target bit rate correspond to key frames which contain a larger amount of data to be transmitted than the other frames.

2.1.4.2 Network Jitter

In this text, we understand network jitter as an expression of the variation in the packet delays. The delays in the packet delivery are causing synchronization problems at the receiver with the streams originally sent at constant bit rates. Consider a sender sending a stream at constant bit rate with constant packet spacing in time. Due to network jitter, the differences in arrival times of the packets at the receiver are not constant any more and therefore the data received per unit time varies and the stream looks more like a variable bit rate stream at the receiver.

Because the digital video data has to be rendered at a constant rate, the receiver has to absorb the network jitter in the incoming stream. The receiver also needs to synchronize its output rate to the rate of the original video stream arriving to the sender in order to produce the original synchronous stream and to avoid buffer underflows and overflows. This issue is addressed by play-out buffers and clock recovery methods described in the following section.

The network jitter can be expressed e.g. as a variance of the inter-arrival times [1] or a sample standard deviation of the inter-arrival times (square root of the variance) and is defined by the following equation:

$$s_P = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (it_i - \overline{it_N})^2}, \quad (2.3)$$

where:

- N : is the number of packets in the sample
- it_i : is the inter-arrival time between packets i and $i - 1$
- \overline{it}_N : is the average of packet inter-arrival times

2.1.5 Playout Problem

The variation in network delay (jitter), means that the time difference between transmitting any two packets at the sender is unlikely to be the same as the difference in arrival times at the receiver. Received packets contain video data that has to be played at a constant rate in order to produce a synchronous video stream. If the packets are played out (rendered) as they arrive, there will be glitches in the output stream because of the packets which arrive later than their expected playout-time. In the worst possible case, a large delay of one packet can cause that all subsequent packets miss their scheduled playout-times and the whole frame has to be discarded. This issue is one of the play-out problems and is solved by a playout buffer as is showed in Figure 2.5.

The playout buffer attempts to absorb the jitter introduced by a network prior to rendering the data to user. It is obvious that a big enough play-out buffer can compensate any amount of jitter. In the extreme case, a buffer as big as the entire video would eliminate any possible jitter at the cost of a large end-to-end delay equal to the transmission time. On the other hand, if the buffer is small then some packets will still arrive too late and will be discarded, causing glitches and gaps in the stream.

The size of a playout buffer is controlled by a playout buffer algorithm, where a good algorithm will attempt to achieve the best trade off between packet loss and delay while keeping jitter low.

Common algorithms for managing a play out buffer follow one of two basic approaches [30]:

Fixed approaches

Fixed approaches assume that the range of delays is predictable and use a static buffer size and schedule. This approach is illustrated at the top of Figure 2.5. The receiver has a large receiving buffer. When buffer occupancy reaches the (fixed) size that is expected to compensate the network jitter, the receiver starts playout. When the media data rate is fixed, the amount of data for jitter reduction is controlled by the network jitter. If the rate is not fixed, the amount of data for jitter reduction is variable. The buffer delay is constant during the transmission.

Clearly, with longer playout delay the probability of late packets arrival decreases. Excessively long playout delays, however, have a significantly impact on the quality of transmission from the human perspective. It has been shown that fixed buffer approaches are unable to achieve satisfactory quality. In particular, they are unable to achieve the best trade off between packet loss and playout delay while keeping jitter low. Therefore, a reactive approach with a self-adaptive system is required [31].

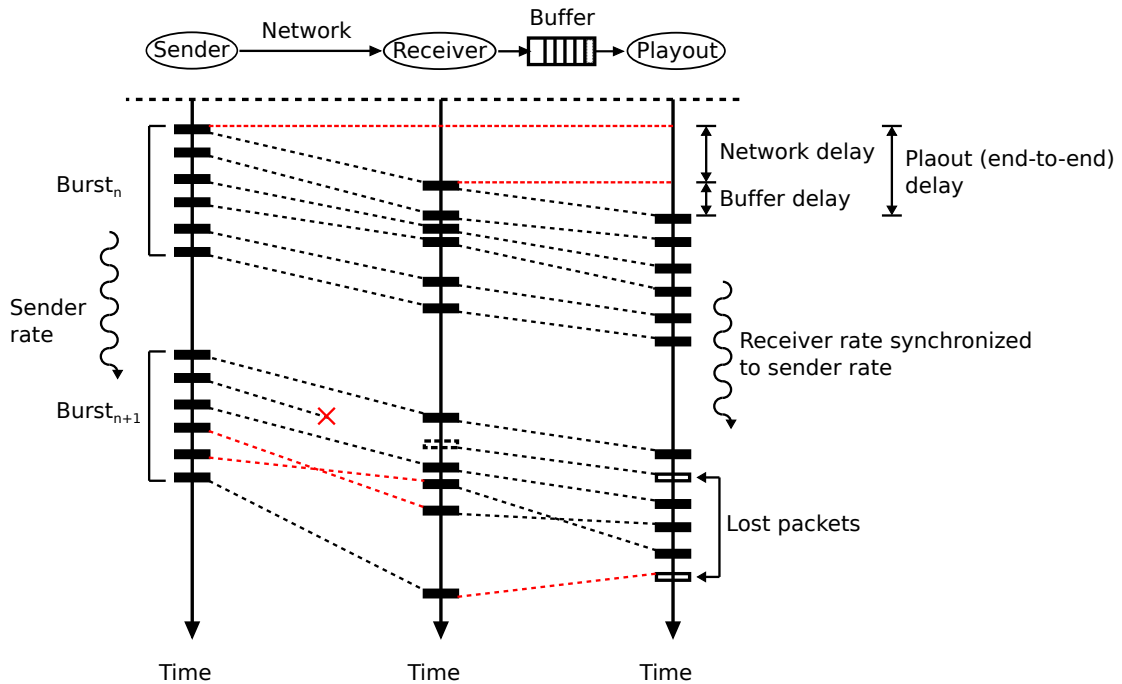


Figure 2.5: The Playout Problem. The black horizontal bars indicate the arrival times of packets at different locations between sender and the playout device. The dashed lines correspond to data packets. The red cross illustrates a packet which was lost in the network. There are two packets which arrived out of order and they have to be swapped. The last (bottommost) packet suffers from excessive delay in the network and arrives later than the expected playout time minus the buffer delay meaning that it can not be display in time and must therefore dropped at the receiver.

Reactive approaches

Reactive approaches measure immediate jitter and use that to dynamically adjust the buffer size and schedule to avoid glitches and gaps so that the percentage of late packets is kept low, typically well under 1% [32].

Various algorithms have been studied in literature for the purpose of the playout buffer delay adjustment. The most common approaches use an autoregressive average of the network delay in the selection of the total end-to-end delay (ted). These algorithms were first used in audio conference tools minimize latency while keeping packet loss due to network jitter at a very low level.

The most classical algorithm is based on the RFC793 (TCP, Transmission Control Protocol Specification) algorithm [33]. The delay estimates are calculated by the method described in [33] and a measure of the variation in the delays is calculated by Van Jacobson's method [34] used for calculation of round-trip-time estimate for the TCP retransmit timer. Note the similarity between the calculation of the optimal value for the TCP retransmit

timer and the calculation of the optimal delay for the playout buffer. Both ideas are based on the same principle: to find a minimal latency with a minimal packet loss. The playout algorithm is presented here as is described in [32].

End-to-end delay for a packet is equal to the accumulated network delay plus buffer delay. The delay estimate \hat{d}_i for packet i is calculated as:

$$\hat{d}_i = \alpha \hat{d}_{i-1} + (1 - \alpha)n_i \quad (2.4)$$

and the variation is calculated as:

$$\hat{v}_i = \alpha \hat{v}_{i-1} + (1 - \alpha)|\hat{d}_i - n_i|, \quad (2.5)$$

where \hat{v}_i is an estimate of the variation in network delay, n_i is the actual network delay for packet i , and α is a weighting factor that controls the rate of convergence of the algorithm. Given these values the *ted* value for playout is calculated using:

$$ted = \hat{d}_i + \beta \hat{v}_i, \quad (2.6)$$

where β is a factor chosen to accommodate changes in network conditions that occur suddenly.

This algorithm is a linear recursive filter and is characterized by the weighing factor α . An optimal settings for α and β is discussed in [35]. For voice streams using silence suppression it is common to adjust *ted* only at the start of a talk spurt, thereby minimizing the impact on audio quality of delay changes. In other situations *ted* can be adjusted periodically, or based on a threshold difference between target and current *ted*.

There are several other algorithms for playout buffer adjustments. An adaptive algorithm based on the Normalized Least Mean Squares (NLMS) filter is proposed in [32], where the playout delay is calculated and updated on every packet. This allows to decrease buffer delay and latency at the expense of slight increase in the number of delayed (discarded) packets.

Another method of network delay prediction is proposed in [31]. It maintains a statistical representation (histogram) of network delays observed during the fixed number of packets, currently 10000, in order to improve the prediction accuracy.

A hybrid solution is proposed in [36], where the classical (TCP) algorithm or a variant is used during the initial phase of the stream and a histogram based algorithm is used later.

An algorithm which measures variations in receiver buffer lengths to determine the playout delay rather than trying to estimate network delays is presented in [37].

2.1.6 Clock Recovery

As mentioned in the previous sections the video transmission problem can be seen as a conversion of a synchronous video stream (continuous video signal) to an asynchronous packet stream and back to a synchronous video stream at the receiver.

Such conversion from synchronous and asynchronous streams appear also in telecommunications, where time-division multiplexing (TDM) circuits are emulated over packet switched network such as TDM over IP (TDMoIP) [2]. TDM is synchronous continuous bit stream at constant bit rate carrying e.g. voice communications offering low delay for real time services. When transmitting TDM data over packet switched networks the original synchronous TDM stream has to be reconstructed at the receiver due to variable network delays. The jitter buffer is used at the receiver to compensate the packet jitter coming from network and a clock recovery method is used to keep sender and receiver clocks synchronized.

Clock recovery methods can be categorized into synchronous and asynchronous.

2.1.6.1 Synchronous Clock Recovery

Synchronous clock recovery methods [38] require the underlying network to have a reference timing source. The sender encodes information on the frequency difference between the source clock and the reference clock into the outgoing packet stream. This difference is calculated using a common network clock available to both the sender and the receiver. The receiver can recover the original source clock frequency using this frequency difference information and the same reference clock.

2.1.6.2 Asynchronous Clock Recovery

Asynchronous clock recovery allows the receiver to synchronize to the transmitter's clock when no common clock source is available to both. This is also called adaptive clock recovery. In this case, the receiver has to derive an estimate of the source clock from the incoming packet stream as shown in Figure 2.6.

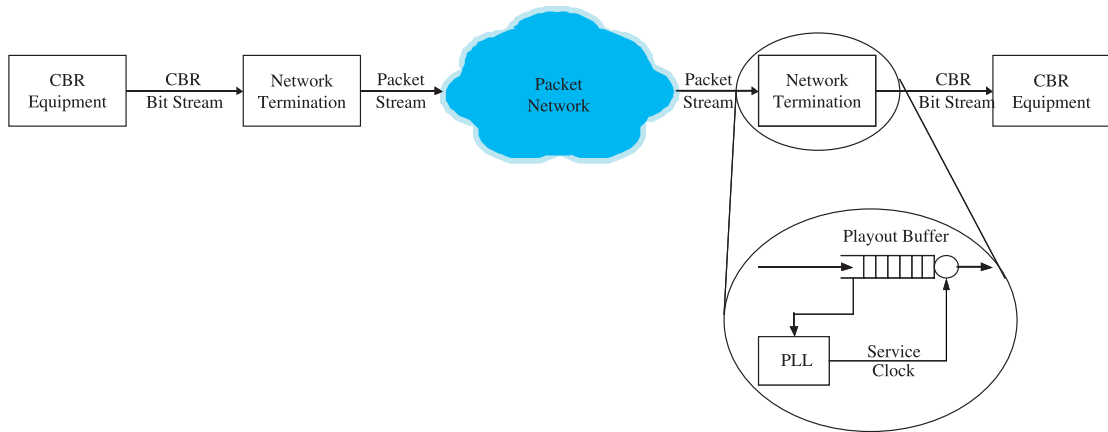


Figure 2.6: Clock recovery block diagram [2]

The clock information is derived from the packet arrival patterns such as the buffer level (occupancy) [39, 40, 41] or the packet inter-arrival times [2] by applying an appropriate

digital filter. Both methods then drives the phase-locked loop (PLL) to regenerate the source clock frequency. The purpose of the PLL is to estimate and compensate for the frequency drift occurring between the sender's and receiver's clock oscillators, the block diagram is depicted in Figure 2.7.

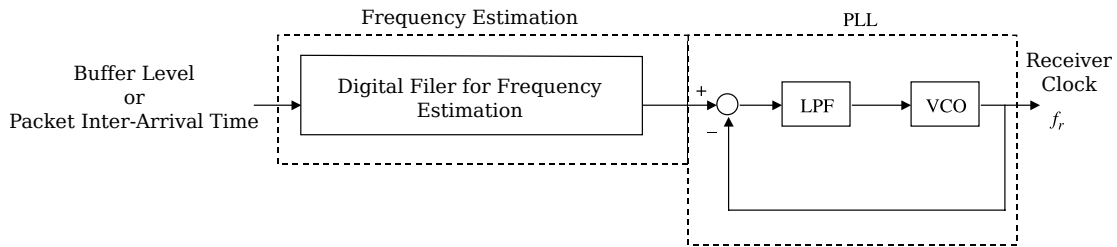


Figure 2.7: Adaptive clock recovery

The buffer level method is based on a (de)jitter buffer (FIFO) and works as follows [41, 42]: It takes the time-average of the observed buffer state for use in reducing the packet jitter and estimating the frequency difference between the source and the receiver clocks, which is then used for controlling the receiver clock frequency. To understand the control mechanism let us assume that the receiver local clock is initially at lower frequency than the source clock. Writing into the buffer occurs faster than it is read and thus the fill-level starts to rise. The conventional mechanism defines zones around the buffer center, and each zone is interpreted as a frequency offset with respect to the nominal local clock. When the buffer level reaches a zone the local clock is set to the frequency associated to that zone. If this frequency now matches the source clock, the buffer level will now remain constant since the filling and emptying are proceeding at the same rate, but if the local clock is still lower in frequency the level will continue to rise until it enters the next zone. Once the level is in the next higher zone the local clock is set to a yet higher frequency, and the process continues until the local clock precisely matches the source clock. Note that this corresponds to a control loop, whereby the fill-level of the buffer causes a change in the local clock frequency, and this clock frequency directly influences the position of the buffer fill-level.

An alternative to a buffer level method is a method based on packet inter-arrival time measurement [2]. Here, a PLL locks onto the source clock frequency estimated from time-averaged packet inter-arrival times. The PLL compensates the frequency difference between the sender and receiver, but without considering the clock phases. Thus, an arbitrary time offset may remain. The main advantage of this method is that it allows to quickly compensate the frequency difference and to retain low frequency wander. The disadvantage is that it can not control the buffer level, and thus does not guard against buffer underflow or overflow [42].

2.1.6.3 Notes on video Synchronization

The clock recovery methods described in the previous section are based on the assumption that only CBR traffic is carried in fixed size packets over packet network and packet losses are assumed to be low.

The full HD-SDI streams or raw video streams (PCM data/digitized samples without any framing) exhibit this property and can be carried as CBR video streams over the packet switched network and the described clock recovery methods can be used to synchronize clock at the two ends, assuming a reasonable jitter and low packet loss. However, such direct HD-SDI or pure raw video transmissions requires higher than necessary network throughput as the blanking intervals not containing any visible data are also transmitted.

When only visible data are transmitted or video encoding/compression algorithm is in place, such a stream does not have a constant bit rate but its speed varies in time. These streams are known as variable bit rate video streams and the described clock recovery methods cannot be used directly as it is difficult to distinguish the variation in the periodicity of the packet stream due to video encoding or compression and due to network jitter [40]. Extra information has to be provided in the data stream so that the clock recovery algorithm or PLL can lock on this information.

2.2 Related Work

This section presents notable related work.

2.2.1 NTT Network Innovation Laboratories

NTT¹ demonstrated the world's first uncompressed 4K video transmission over a long distance network (21000 km) in 2007 [43]. The presented system consists of four server PCs: Two are used on the sender side and two on the receiver side. Each PC is equipped with two HD-SDI I/O boards and a 10 Gb/s Ethernet interface. A 4K Camera is used as a 4K image source and provides four HD-SDI signals connected to the first two PCs. The video signal is captured and encapsulated into four UDP packet streams without any compression. The rate per one UDP stream is 1.5 Gb/s adding up to 6 Gb/s for the entire 4K image. The system on the receiver side converts the received UDP streams into four HD-SDI signals and then creates one master video stream from them. A master video clock is regenerated in the HD-SDI output board from the master video stream. The other three HD-SDI interfaces use this master clock as a reference clock and synchronize their receive buffers to it. The exact type of clock regeneration and synchronization was not described. Furthermore, the frame processing running on the different PCs have to be synchronized in order to establish video frame synchronization between the different HD-SDI signals. For this purpose, the two PCs are configured to exchange the current playing video frame number using another Ethernet port. The exact processing delay was not presented. This delay is has to take into account HD-SDI frame input delay, software processing time, frame synchronization latency and HD-SDI frame output. The HD-SDI input/output to/from PC is typically through the frame buffer, thus, the latency is probably more than the length of single frame (30 msec), or in its multiples depending on how many frames are buffered.

NTT also demonstrated the first compressed 4K video transmission over a long distance (15000 km) with hardware accelerated JPEG 2000 compression implemented in a custom PCI-Express board [44, 45]. The presented system consists of two server PCs, one for each transmission end-point. Each PC is equipped with the four of these PCI-Express custom boards. The custom board contains two JPEG 2000 encoder/decoder based on the ADV202 chip from Analog Devices and two HD-SDI input and two HD-SDI output interfaces. The board also has a clock synchronization input. The JPEG 2000 compression reduced the network stream rate three times down to 500 Mb/s. The delay generated by JPEG 2000 encoding, transmitting and decoding is up to 6 frames in total, which equals to 165–198 msec.

¹NTT Network Innovation Laboratories is research and development center of Nippon Telegraph and Telephone Corporation (NTT) which is a Japanese telecommunications company headquartered in Tokyo, Japan.

2.2.2 UltraGrid

UltraGrid² [46] is a software technology enabling low-latency transmission of uncompressed or very low-compressed high-definition video in HD, 2K and 4K resolutions. UltraGrid is used, among others, in areas like collaborative environments, medical cinematography, broadcasting applications and various educational activities. UltraGrid consists of a software solution running on PC with Linux or MacOS and a 10 Gb/s network connection optimized for low latency applications. The PC is equipped with recommended professional audio/video hardware devices including HD-SDI I/O boards and CUDA based graphics card for video compression/decompression acceleration.

UltraGrid achieved 175 msec [47] end-to-end latency in a laboratory set-up with two computers equipped with a HD camera, an HD-SDI interface card, HD monitor. The computers were connected to a CISCO switch using a 10 Gb/s local network through the CISCO switch. The quoted latency includes the camera processing delay, grabber delay, software processing and LCD response time. The processing delay (excluding the camera and LCD delay) is 122 msec. The image resolution was interlaced 1920x1080 at 60 frames/sec. The receiver to sender synchronization is frame based, thus the processing delay corresponds to several frame durations. The synchronization uses the time information from the RTP/RTCP packets. When MPEG-2 based compression is enabled, the end-to-end latency increases to 1907 msec.

2.2.3 LOLA – LOW LATency audio visual streaming system

The LOLA project³ [48] aims to enable real time musical performances where musicians are physically located in multiple remote sites, connected by advanced network services, like the ones provided by the NRENs and GEANT and other International backbones. LOLA sends uncompressed audio/video data, grabbing them from the hardware and sending to the remote site over the network totally unprocessed. The LOLA system consists of a software solution running on PC with Microsoft Windows and 1 Gb/s network connection optimized for low latency applications. The PC is equipped with recommended audio/video hardware devices including an audio I/O board, video grabber board and monitor. Each device is optimized in order to achieve the lowest possible latency, the monitor is selected based on the pixel response time. The camera and audio control software are rewritten from scratch and the networking packet drivers were also greatly re-engineered.

The LOLA project achieved 30 msec end-to-end latency [49]. The round trip latency is about 60 msec from instrument to human ear and back fooling the ear into believing that the musicians are in the same room. The current supported resolutions are up to 640x480 RGB at 60 frames/sec with 2x 44100 Hz 24bit audio. This requires throughput over

²UltraGrid is open-source software, distributed under the BSD license. UltraGrid is developed by the Laboratory of Advanced Networking Technologies (SITOLA, also known as ANTLab) and supported by CESNET association.

³ LOLA is a project developed by Conservatorio di Musica Giuseppe Tartini from Trieste (Italy) in collaboration with GARR, the Italian Research and Academic Network.

500 Mb/s. As described in [50], LOLA does not implement any data loss recovery feature to save time, thus the network must provide a very reliable zero packet loss performance for hours. LOLA does not buffer audio/video information (even if it can use some buffers, to adapt to less optimal conditions), the network delay must be very stable and therefore the jitter small. The network jitter requirement is for jitter less than 3ms at 30 frames/sec and less than 6ms at 60 frames/sec.

2.2.4 Commercial devices

The intoPIX company offers a similar streaming solution like NTT does. It consists of a PCI-Express board with a JPEG 2000 coder/decoder implemented in the FPGA and a server PC for RTP over IP network streaming with a processing delay of 3 frames (125 msec) including encoding and decoding time [51]. However, the presented solutions do not provide frame synchronization which can only be achieved by providing an external clock signal to sender and receiver such as studio-wide clock source or local genlock generator.

Net Insights Nimbra 600 [52] series of switches can transport 8x HD-SDI or 3G SDI channels over a SONET/SDH network using a video interface line card with optional JPEG 2000 compression. However, the frame/clock synchronization and latency are not described in the available documentation.

2.2.5 Summary

Each of the discussed systems has a HD-SDI video processing PCI-Express card which is used to connect to the video source. This card can either be a commercial HD-SDI interface card or a custom built card with a hardware compression engine reducing the data rate. Then a server PC is used for network transmission, reception and frame synchronization (the intoPIX system uses an external clock source). The processing delay is equal to or larger than one frame time (30 msec). The only exception are Insight's Nimbra devices which integrate the video transport with the network device and are thus similar to our architecture. However, Insight doesn't disclose any details about the receiver to sender synchronization and processing latency, therefore we can not compare with their architecture.

The architecture presented in this doctoral thesis is a new approach and as such there are only a few comparable solutions. The main advantage of the presented technique over the existing solutions is that it makes use of small memory buffers (much less than the buffer required for one image frame), thus adding only a minimum amount of latency. Another advantage is that the presented architecture is based on the 10 Gb/s network modular processing platform briefly described in the following Section. Thus, it does not require any additional server PC for network connection and processing. It is a single and portable FPGA based device designed for a maximal data throughput. The 10 Gb/s network modular processing platform is described in the following section as my previous results.

We presented a first working prototype during the CineGrid 2009 workshop [53]. To our best knowledge, our prototype was the only device capable of bi-directional transmission of 8x HD-SDI signals with processing delay less than 1 msec, with receiver to sender clock synchronization and frame synchronization between the streams.

2.3 My Previous Results

This section briefly discusses my previous results achieved in the area of high-speed passive network monitoring upon which the work presented in this thesis builds. I contributed to the research of hardware accelerated passive network monitoring. The main achievements are described in the following sections. To a great extent, these enabled the research in the low-latency high-resolution transmissions described in the next chapter.

2.3.1 Background

Network monitoring can be divided into the two distinct categories:

Active monitoring – Uses artificially injected test traffic (probes). This is the simplest monitoring technique, for example it allows direct measurement of traffic delay and its variations (jitter). However, the injected traffic can influence traffic already present on the measured medium and cannot be used to detect some properties present only in real user traffic.

Passive monitoring – Is monitoring purely based on measurements of existing traffic. In contrast to active monitoring, typically all user traffic is captured and processed. Therefore, additional traffic properties can be measured such as time-dependent characteristics, network attacks, real packet loss rate, etc.

Capturing and processing incoming traffic for passive monitoring is a difficult task as it requires to capture and process all input traffic regardless of packet size. In [A14, A15] we showed that the standard 10 Gb/s network cards are inappropriate for this task due to the high packet loss for short packets (up to 90% of traffic is lost) and we proved that a specialized monitoring device is required. These devices are usually based on a FPGA.

Field Programmable Gate Arrays (FPGAs) [54] are semiconductor devices designed to be configured after manufacturing (“field-programmable”). The majority of current FPGAs are based on static memory technology (SRAM) and are re-programmable. FPGA devices consist of large number of configurable logic blocks connected via programmable interconnects. The logic blocks can be configured to perform complex combinational functions or to act as flip-flops. High-end FPGAs also contain more memory elements, high-speed serial I/O links, DSP blocks and embedded processors.

2.3.2 Modular Traffic Processing Platform (MTPP10)

The Modular Traffic Processing Platform (MTPP) is an FPGA based device for for high speed packet monitoring at speeds 10Gb/s and higher. The platform can receive, analyze, modify or generate packets. The requested functionality is implemented in plug-in modules which are loaded at the run-time and can be changed later without the device interruption.

The platform architecture is shown in Figure 2.8. It consists of a FPGA based hardware accelerator utilising number of reconfigurable plug-in modules and an embedded processor.

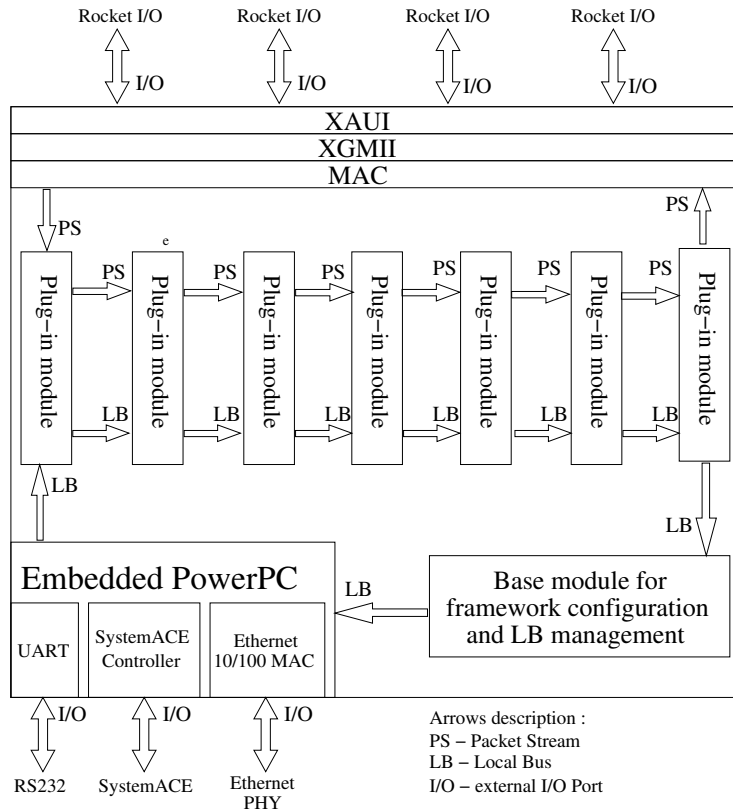


Figure 2.8: Modular Traffic Processing Platform Architecture

The hardware accelerator manages all time critical packet operations, while the embedded processor manages all non-time critical operations such as configuration and statistics collection. The modules are loaded by the FPGA dynamic partial reconfiguration [55] where only a subset of the FPGA is modified. The rest of the FPGA remains operational and unaffected. Partial reconfiguration allows us to use many different hardware modules with a wide area of functions that can be freely combined and assembled to build various final functions without full device reimplementatation.

This result was achieved in CESNET together with my colleagues Dr. Sven Ubik and Jiří Halák. The main processing core was proposed by Jiří Halák and is designed to be fully scalable to 10/40/100Gb speeds. It is described in more detail in his dissertation thesis.

My contributions to this platform include: A new framework for dynamic partial reconfiguration simplifying the FPGA reconfiguration process, several reconfigurable and static modules and software libraries. In particular, designed and developed the `pkt_loss` module (packet loss emulation for protocol testing) and `packet_gen` module (packet generator supporting emulation of various kinds of network protocols), a 10 gigabit network interface for the embedded processor and a complete software package including system

management tools and a specialized embedded Linux distribution for PowerPC CPU and later Microblaze CPU directly synthesized in the FPGA. For the platform I also designed an oscillator board which provides a precise and temperature compensated clock for the optical transceiver and the FPGA.

The modular traffic processing platform is published in [A13] and presented by Jiri Halak in [56, 57]. The platform is protected by Czech patent [A25] and by utility patent [A28]. We designed a platform prototype [A31] shown in Appendix A. The platform is under routine operation in the CESNET2 network.

2.3.3 Modular Traffic Processing Platform 40 Gb/s (MTPP40)

With the help of my colleague Dr. Sven Ubik I successfully designed and developed a prototype platform for testing 40 Gb/s networks – the Modular Traffic Processing Platform 40 (MTPP40). This research started in 2008 as a feasibility study of the development of a such high-speed device with a commodity FPGA. The CESNET optical department showed strong interest in such a device as 40 Gb/s links were used more and more.

The first project based on this platform was a bit error rate tester (BERT) for testing 40 Gb/s SONET/SDH network circuits and for evaluation of properties of optical transceivers over long-distance links. The BERT was designed according to ITUT recommendation O.150 (Digital test patterns for performance measurements on digital transmission equipment) [58].

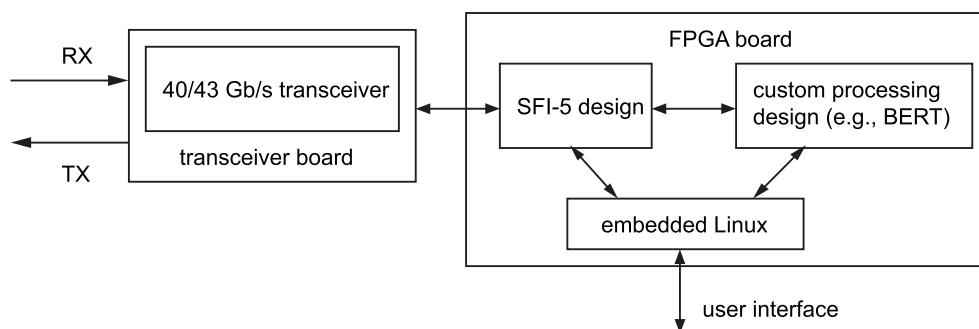


Figure 2.9: Hardware Architecture of the 40 Gb/s Modular Traffic Processing Platform

The general hardware architecture is shown in Figure 2.9. We use an Opnext 40 Gb/s optical transceiver connected to the FPGA with 34 high-speed electrical lanes via an interconnecting adapter board. The FPGA implements the SFI-5 interface with lane deskew, BERT module and Microblaze CPU running embedded Linux operating system.

For this device, I successfully designed the interconnecting adapter board with 34 high-speed differential links, the oscillator board which provides the precise and temperature compensated clock signals for the optical transceiver and the FPGA, BERT FPGA firmware capable of generating and validating bit patterns at 40 Gbps and a specialized embedded Linux distribution for the Microblaze CPU synthesized in the FPGA.

The 40 Gb/s modular traffic processing platform is published and presented in [A12]. We designed the following platform prototype [A31] shown in Appendix A. I presented the platform prototype in operation at the Field Programmable Logic and Applications conference (FPL) 2009 in Prague. The MTPP-40 platform is used by the CESNET optical department for bit error rate tests (BERT) of optical fibres and network paths in experiments of long-distance all-optical transmissions.

2.3.4 Summary

In this section we briefly discussed the results achieved in the area of high-speed passive network monitoring. This area was studied intensively by the CESNET research teams [59] as one of the CESNET's main goals. We developed a set of hardware devices and software tools able to process network traffic at speeds of 10 Gb/s and higher.

We took experience gained from the network monitoring research and applied them later to the research in the area of high-speed low-latency video transmissions where we identified the future applications of high-speed networks. The main contributions in this area are described in the following chapter.

Chapter 3

Contributions

This chapter discusses the main contributions of this doctoral thesis. Every section in this chapter presents one contribution. The selected published results are available in chapter 5.

3.1 A technique for receiver synchronization in video streaming with short latency over asynchronous networks

3.1.1 Introduction

I proposed a new and now patented technique for the receiver to sender clock synchronization based on a measurement of the receiver's processing delay. The main advantage of the technique is that it makes use of small memory buffers (much less than the buffer required for one image frame), thus adding a minimum latency.

This technique is based on the asynchronous clock recovery described in Section 2.1.6 and measures the receiver's processing delay, i.e., the elapsed time between receiving the video data from a network and sending out to the rendering device. The elapsed time includes the time spent in the input buffer. A new approach used in this technique is the usage of the processing delay for frame alignment adjustments (vertical synchronization). This approach has several benefits: it guarantee the frame alignment, controls the receiver buffer occupancy, it is network jitter resilient and requires only small memory buffers. Thus, adding a minimal latency which is very important for transmissions for remote collaboration.

To overcome the issues of receiver to sender clock synchronization on low-latency devices with small memory buffers we investigated several possible approaches including: receiver feedback to the sender, frame buffer, blank period adjustments, external clock synchronization and rendering clock adjustments. The following paragraphs briefly summarize each approach:

Receiver feedback The receiver can send feedback to the sender requesting sending rate adjustments (flow control). This technique is used in window-based transport layer protocols, such as TCP or in some link layer protocols, such as PAUSE frames in Ethernet. The data with the adjusted data rate arrive at the receiver after one round-trip time (RTT), which is typically hundreds of milliseconds on long-distance Internet connections. In high-definition video transfers, this technique would require a receiver buffer with the size equal to the RTT, which would introduce additional latency.

The advantage of this approach is that it can guarantee a reliable delivery. The lost data are retransmitted. The disadvantage is the high latency due to the use of larger buffers. Therefore we decided to use another technique.

Frame buffer This technique requires the receiver to have a buffer memory sufficiently large to store at least two complete image frames (a so-called frame buffer). The rendering starts when the first full frame is loaded into the buffer. This allows the rendering device to be driven by a fixed clock source in the receiver. The buffer works as a FIFO memory, the top of the buffer holds the first frame that is being sent to the rendering device first.

After a long period when the skew between the sender and receiver clock rates causes the next frame to be rendered to not be completely available in time, the previous frame is rendered again. Similarly, if the buffer cannot accommodate more incoming data, one frame is dropped. Of course, this solution will cause a disruption to the audio signal if the audio signal is transferred along with the video signal in one stream.

We did not use this option because it would require an external memories to be used with an FPGA and would introduce high latency and problems with audio synchronization.

Blank period adjustments Most digital video transfer formats include a blank period in addition to an active period (visible lines) in each frame. The blank period is used for embedded audio, ancillary data and synchronization.

An interesting technique of adjusting a rendering speed is to add or remove some samples in the blanking periods. This technique is in compliance with SMPTE-274M, Annex E.2 [60], where the blanking periods can be moved up to 6 clock periods. The advantage of this technique is that the clock synchronization doesn't require any external hardware components (e.g. tunable clock oscillators). Therefore, it can be directly implemented in an FPGA. We found that it works perfectly with our laboratory devices but it proved to be fatal with some of the Cinema-grade projectors displaying a blank image for several seconds.

Precise clock synchronization When both the sender and receiver are driven by a precise external frequency source, such as one locked on a PPS (Pulse Per Second) signal from a GPS (Global Positioning System) receiver, the whole system can run stably for a long time. However, some senders cannot be synchronized to an external frequency source,

such as medical equipment. Moreover, it is often difficult to obtain GPS signal in locations where video signals need be to sent and received, for example in lecture halls.

Rendering clock adjustments Adjusting the clock of HD-SDI channels between the receiver and the rendering device within the permitted tolerance (± 10 ppm [60]) gives the receiver some level of adaptation to the rate of incoming data. This solution requires a tunable oscillator and a closed-loop controller in the receiver. My proposed technique is based on this type of synchronization. The commonly used techniques were discussed in chapter 2.1.6.

3.1.2 Architecture Overview

The synchronization block diagram is shown in Figure 3.1. Video data arriving from a network are stored in the input buffer (FIFO memory). The FIFO acts as the playout buffer as was described in chapter 2.1.5. The FIFO reduces network jitter and synchronizes the data to the rendering clock at which the data are read, further processed and rendered. The rendering clock is generated by the tunable clock generator and fine-tuned by the PID controller. Consequently, the regulation system can control the input buffer occupancy and latency by small adjustments in the rendering speed.

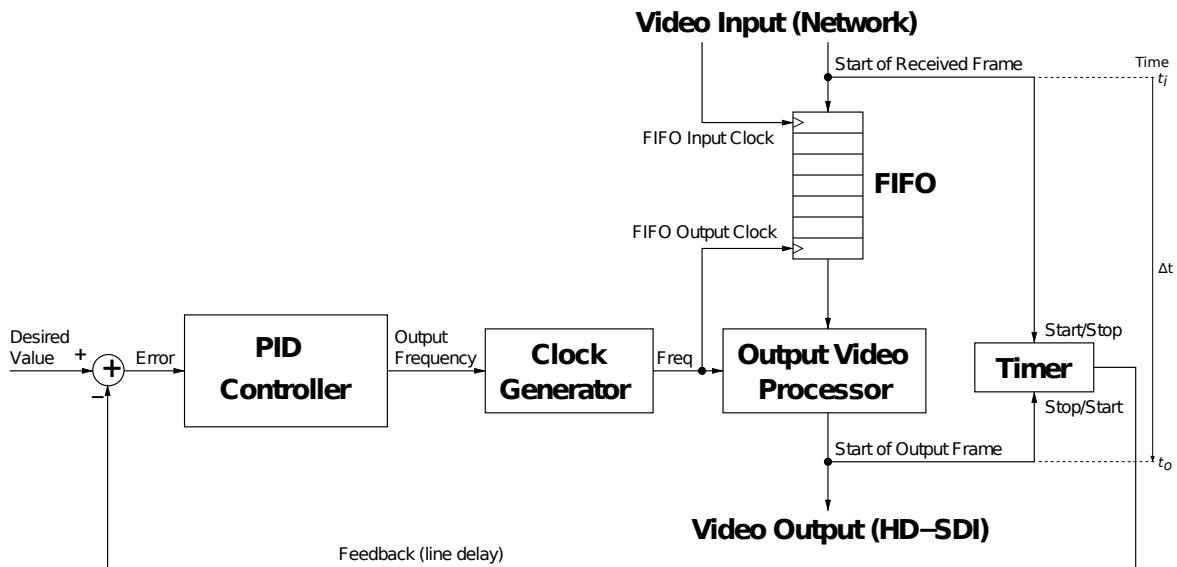


Figure 3.1: Synchronization block diagram

In Figure 3.1 the feedback signal is the result of delay measurement between the time when the first active line (start of the received frame) enters the receiver FIFO memory and the time when the output video processor sends the first active video line (start of the output frame) to the rendering device. The delay is measured in the clock cycles. The feedback variable is compared to the desired value (desired delay), whose value is

empirically optimized for maximum picture stability. This is similar to the fixed playout approach discussed in Section 2.1.5. The PID controller produces the adjustments to the clock generator in order to keep zero difference between the feedback and the desired value. Consequently, the processing delay (the feedback value) is constant and on average the rendering clock is equal to the clock of the sender.

A bare adjusting of the rendering clock frequency based on e.g. buffer level feedback (see Section 2.1.6.2) would temporarily stabilize the rendering clock frequency, but there are several remaining issues (an explanations is below):

- As was discussed in Section 2.1.6.3 this method would fail when a variable bit stream is received (e.g. video signal with blanking areas removed).
- The buffer control is lost during the buffer underflow.
- Without any additional logic it doesn't guarantee the frame alignment between the lines received from the stream and the lines send to the rendering device, i.e., the top of the picture could appear anywhere on the screen.

The loss of the frame alignment depicted in Figure 3.2. The two upper quadrants of the 4K image seems to be shifted up by one line causing the appearance of a white line in the middle of the screen. In fact, the all four quadrants are shifted up causing one missing line at the bottom of the quadrants.



Figure 3.2: An example of receiver synchronization without frame alignment demonstrated with Sony SRX 4K projector. (Note the white line in the middle of the screen.)

The proposed method guarantees frame alignment and buffer control:

- Let t_i be the time when one active line with an arbitrary line number enters the receiver (input time). For the sake of simplicity let's assume it is the first line, but any allowed line number can be considered.
- Let t_o be the time when the line with the same line number must be sent to the rendering device (output time), which is not necessarily the input line. It will be an artificially created line if the input line is not available.
- Let $\Delta t = t_i - t_o$ be the difference between the input and output time. The output time t_o is controlled by the local clock generator and is independent of the input time. The controller tries to keep Δt stable and thus keeping the local clock synchronous to the clock of the sender. Therefore, Δt is used as the feedback value for the controller and the desired value is the requested (expected) time difference.
- Let t_{frame} be the time required for rendering one video frame, which is the inverse of the frame frequency.

We consider three cases:

1. When $t_i < t_o$ (the line arrives at the input before it should be sent to the rendering device), the input line waits for a period $|\Delta t|$ seconds in the input buffer before it is rendered at time t_o .

Because of the network jitter, the line can be delayed on the input. In the next frame, the line with the same number can arrive at the input with delay up to $|\Delta t|$ with respect to the projected arrival time $t_i + t_{frame}$.

2. When $t_i \geq t_o$ (the line arrives after it should have been sent to the rendering device), the input line is delayed and cannot be processed before the rendering time. Therefore, the line is dropped at the input. This situation is also equal to an empty input buffer. Thus, at the rendering time t_o a new line is artificially created to fill the gap and is sent to the rendering device. This new line can be a blank line. However, during the field tests we found that the best is to send a copy of the previous line. A copied line is much harder to notice than a blank line because it matches the color of the neighbouring lines.
3. When the input buffer is full, newly arriving lines must be dropped at the input. The rendering continues with the available lines causing the temporal shifting of the image up. The dropped lines are missing at the bottom of the image. That part of the image is filled with the new artificially created lines.

3.1.3 Summary

I proposed a new technique for the receiver synchronization based on a measurement of the receiver processing delay. The presented synchronization technique has several features:

- Synchronization doesn't depend on the packet size – A single line can consist of one or several packets. Usually, the line video data has to be decapsulated or decompressed from the packet data. This process is transparent to the synchronization, because synchronization is based on the time measurements of line input and output events. Thus, the measurement doesn't depend on the individual packet rate. This is particularly useful for variable bit rate streams.
- Control of buffer occupancy – Increasing or decreasing the desired negative Δt value controls the amount of data stored in the input buffer. During $\Delta t < 0$ the line arrives at the input before it should be sent to the rendering device. Setting a large negative Δt causes arrival of several lines into the input buffer. By setting appropriate desired value the input buffer can be filled to e.g. to 50% of the capacity increasing the network jitter resilience.
- Control is not lost during input buffer underflow – Measuring the difference between the arrival time and the required playout time of a certain active line in the proposed method allows the control circuit to know by how much the buffer underflows and therefore the regulator responds proportionally by decreasing the clock frequency. This is unlike the buffer level based clock synchronization, where it is not known by how much the buffer did underflow. There are three main causes of buffer underflow:
 - The receiver rendering clock frequency is higher than the sender clock frequency (after power up, changing the video source, etc).
 - The lines are delayed on the input due to the network jitter.
 - When a variable bit rate stream is received. For instance, during the vertical blanking periods no lines are received causing the buffer underflow (depending on the input buffer size and the length of the blanking interval). The proposed method counts only active lines, therefore, it is not influenced by blanking intervals. As opposed to the buffer level based clock synchronization.
- Guaranteed frame alignment – As was discussed in Section 2.1.6.2, the clock recovery techniques compensate the frequency differences between the sender and receiver, but without considering the clock phases. Thus, vertical (frame) synchronization is lost and cannot be restored. The proposed method tries to keep Δt stable, thus, is also keeping stable time offset (phase) between the input line and rendered line. Therefore, the input line is rendered at the correct position. The frames are vertically synchronized.

The proposed synchronization technique is published and presented in [A2, A1] which are included in Chapter 5. The technique was experimentally verified and subsequently tested in operation [A7, A9] with the FPGA based device MVTP-4K (Modular Video Transfer Platform) [10]. The technique is protected by Czech utility patents [A26, A22], Czech patent [A24] and international patent [A23] which is also included in Chapter 5.

3.2 Proof-of-concept prototype for low latency video high-resolution transmission

3.2.1 Introduction

The second result of the doctoral thesis is a prototype implementation of the proposed receiver synchronization – the Modular Video Transfer Platform (MVTP-4K). It is an FPGA based extendable and portable platform for full-duplex low-latency high-resolution video transmissions with up to eight video streams with HD, 2K, 4K or 3D 4K resolutions over a 10 Gigabit packet network. The video interface is based on the HD-SDI standard used in professional multimedia devices and is discussed in details in Section 2.1.2.

The proposed platform is based on the extendable modular traffic processing platforms from our complementary research and is described in Section 2.3. The following list summarizes the new approaches used and the results achieved with the prototype implementation:

- The proposed receiver synchronization technique is controlling the small play-out buffer and keeping the receiver synchronized to the sender with a processing delay below 1 msec.
- The video processing logic described in Section 3.2.2.1 and the network traffic processing cores are implemented on the same FPGA. This reduces the inter-communication latency to a minimum because there is no requirement for an external PC and networking software.
- The video processing functionality is split into several blocks each of which is implemented as a separate dynamically reconfigurable extension module. I proposed an architecture simplifying the module preparation and reducing the complexity of the hardware required for the dynamic partial reconfiguration. The architecture is described in Section 3.2.2.3.
- I proposed a new extendable software framework for packet filtering and classification configuration. The framework is able to configure different hardware and different network monitoring cards with classification rules specified in only one language. The framework is described in Section 3.2.2.5.

The prototype was implemented in CESNET together with my colleagues Dr. Sven Ubik and Jiri Halak. The reconfigurable core was proposed by Jiri Halak and is designed to be fully scalable to 10/40/100 Gb speeds and is described in his dissertation thesis. The MVTP architecture is described in the following section. My other contributions to the MVTP include the design of the video interface board, the timing systems, a 10 Gb/s network packet filter, the interface of the networking logic to the Microblaze CPU, and also a complete software package including system management tools and a specialized embedded Linux distribution for Microblaze CPU synthesized in the FPGA.

3.2.2 Architecture Overview

The general hardware architecture is shown in Figure 3.3. The video interface board converts high-speed (1.485 Gb/s) electrical signals between the HD-SDI interface and the FPGA. The tunable oscillator provides the timing. The FPGA board processes the video signal, optionally applies image transformation, compression or decompression and packetizes the video data stream. The FPGA board is connected to the optical transceiver which converts electrical and optical signals for network transmission.

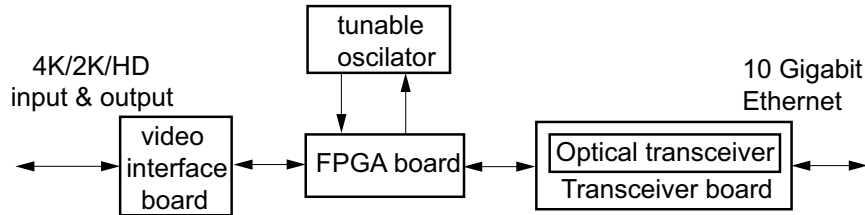


Figure 3.3: Hardware architecture

We selected a Xilinx Virtex 5 FPGA, which allowed us to reuse design blocks that we developed for network monitoring platforms and devices.

3.2.2.1 Firmware architecture

The firmware structure is shown in Figure 3.4. The possibility to partially reconfigure the FPGA by re-loading the modules into the FPGA reduces the time needed for implementation (logic synthesis, placement and routing) of the module by a factor five (from typically up to 60 minutes for the implementation of the entire platform). Since the design is modified very often during the development and testing phase, such a reduction makes the development cycle much more efficient. In addition, having the interface between modules already defined, the developer does not need to spend time to define a new interface and can concentrate on the module's core functionality instead.

The key part of the design is a set of processing modules in the middle and associated switches that can be used to direct video data into different subsets of modules and thus turn the device into different modes of operation (video to/from network processing, video processing only, network processing only).

The function of the core firmware modules is as follows: The input and output MAC implement the link layer and collect packet statistics. The input and output interface attachments serialize the incoming HD-SDI data to the FPGA internal data bus. The encoder and decoder multiplex and demultiplex data from individual HD-SDI streams, insert and extract the active area (visible part) of picture frames and convert four 10 bit wide words on the HD-SDI interface to five 8 bit wide words on the internal bus and vice versa.

The video signal can arrive from video inputs, each connected to a camera or a streaming server, or from the IP network or from both sources. Similarly, the video signal can be sent

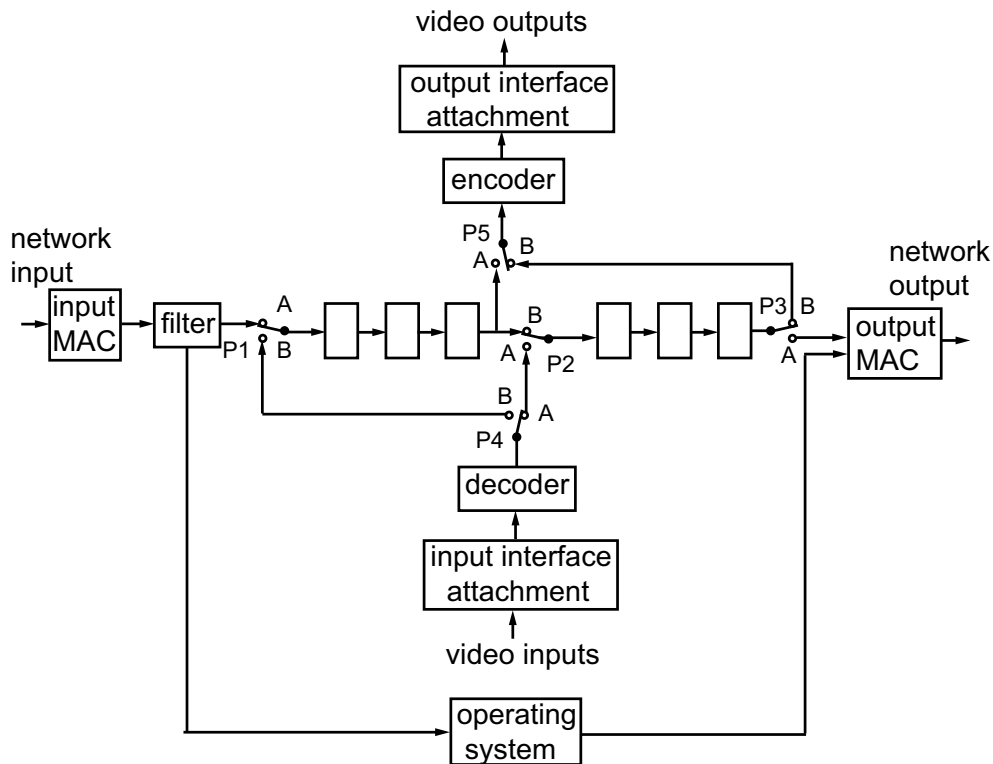


Figure 3.4: Firmware architecture of the prototype implementation

to the video outputs connected to rendering devices or to the IP network or both. The video data are transmitted as a payload of UDP packets. Multiple inputs and outputs can be used together for a higher-resolution picture, such as 4k x 2K. The filter diverts UDP packets containing video into the video processing core and packets of service protocols, such as ARP, ICMP and TCP, to the operating system for further processing. The Linux 2.6 operating system runs on an embedded processor inside the FPGA. I developed two versions — for the hardcore PowerPC processor and for the softcore Microblaze processor.

The receiver synchronization proposed in this doctoral thesis is implemented as described in Figure 3.1 in the previous section. The implementation is divided into two parts. The hardware part providing the processing delay measurements is a part of the output interface attachment. The controller loop is implemented in the software running on the embedded processor inside the FPGA. The controller drives a tunable oscillator which sets the clock frequency of the video processing part and the outgoing HD-SDI channels. Using this mechanism we can stabilize the rendered picture even though there is no connection to the sender clock over an asynchronous IP network.

3.2.2.2 Plug-in module architecture

To allow end users to easily modify hardware functionality without programming, we designed a plug-in module architecture. The packet stream passes through a sequence of modules, each module process the stream and forwards the result of processing to the next module. Typical applications require multiple stages of video processing: When the video signal is transported over an IP network, the first module in a sequence extracts the image lines from the incoming packets. The following modules implement additional processing, such as transcoding, encryption, etc. The last module in a sequence encapsulates the image lines back into the outgoing packet stream.

I designed a specialized embedded Linux distribution for Microblaze CPU synthesized in the FPGA and a set of system management tools configuring the platform and controlling the module reconfiguration. One can use a configuration utility to list loaded modules, to exchange the modules by dynamic reconfiguration, to list registers available in each module and to read from or write to these registers.

```
# mtp -l
Detected modules:
SLOT[0]: "PLATFORM", module v0.2
SLOT[1]: "NIC-RX  ", module v0.1
SLOT[2]: "IP-UNPAC", module v0.2
SLOT[3]: "INIT-MOD", module v0.3
SLOT[4]: "INIT-MOD", module v0.3
SLOT[5]: " IP-PAC ", module v0.2
SLOT[6]: "NIC-TX  ", module v0.3
```

Figure 3.5: List of the currently loaded plug-in modules for video processing. The list is provided by `mtp` tool executed in the embedded Linux system.

For instance, modules listed in Figure 3.5 are part of design for bidirectional 4K video transmissions with 4:2:2 subsampling. The `PLATFORM` module implements the core non-reconfigurable part of the FPGA: various I/O functionality, an interface between the reconfigurable modules, network counters and input and output MAC layer. The `NIC-RX` and `NIX-TX` modules implement the packet filter and 10 Gb/s network interface used for service protocols, which are passed between the network and the embedded Linux kernel. The `IP-UNPAC` and `IP-PAC` modules extract the video data from incoming IP packets and insert the processed video data to outgoing IP packets, respectively. The `INIT-MOD` are dummy modules loaded at startup time before any other modules. They are just passing the packet stream to the next module without any modification.

3.2.2.3 A SystemACE based architecture simplifying partial reconfiguration

The FPGA is configured by downloading a configuration file (bitstream) through the configuration port (JTAG). The configuration process must be repeated each time the device is powered up, because FPGA is volatile. To simplify this step Xilinx developed a SystemACE controller [61] that can configure an FPGA using an ACE files stored in a removable CompactFlash card. The controller is connected to the FPGA through the 16 bit parallel microprocessor interface (MPU) as is shown in Figure 3.6.

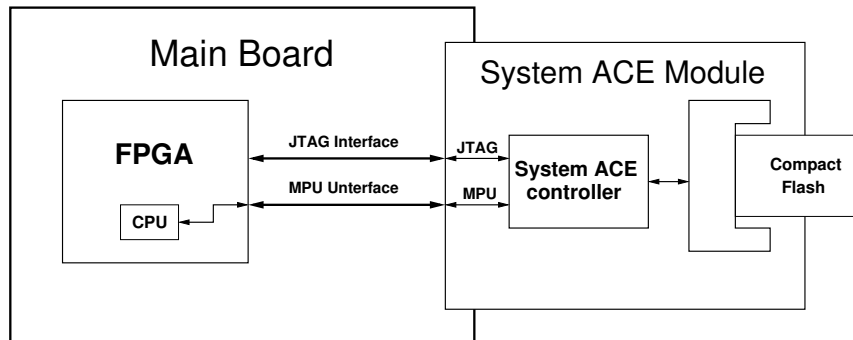


Figure 3.6: FPGA Configuration using System ACE controller

The ACE file is a package compiled from the original bitstream(s) and additional information used to initialize embedded memories and start the embedded CPU. Originally, SystemACE is used only for initial FPGA configuration after the device is switched on. I found that it can also be used to load specific partial ACE files enabling the dynamic partial reconfiguration of an FPGA [57]. This kind of reconfiguration was presented during the FPL conference in 2009 by Jiri Halak and my work is acknowledged in his doctoral thesis. Traditional methods of dynamic reconfiguration are through the FPGA's internal configuration port (ICAP). However, its use is much more complex than the SystemACE interface and can take large amounts of resources [62].

The access to the CompactFlash is provided by the Linux kernel module `xilinx_sysace`. However, this module supports only the read/write access to the CompactFlash but does not allow to trigger a partial reconfiguration of the FPGA from a software application running on the embedded CPU. I added the full support for partial reconfiguration into this driver. The system ACE chip allows to specify up to eight configurations stored on the CompactFlash card. Each configuration has a unique address.

I proposed an architecture simplifying the module preparation and reducing the complexity of the hardware required for the dynamic partial reconfiguration. The architecture consists of three parts:

- A set of scripts based on the FPGA design tools automatically preparing the ACE files from dynamically reconfigurable bitstreams. The reconfigurable bitstreams are produced by standard design tools.

- A set of tools running in the embedded Linux and managing the content of the CompactFlash. When a new configuration of modules is requested these tools automatically rearrange the content of the CompactFlash, create the necessary configuration files and trigger the dynamic reconfiguration.
- A modified `xilinx_sysace` kernel module for embedded Linux providing the dynamic reconfiguration API to the user space through the SystemACE chip.

The partial self-reconfiguration mechanism is one of the most important techniques of the proposed video transmission platform.

3.2.2.4 10 Gb/s network interface

I developed a 10 Gb/s network interface for the embedded processor. The network interface is divided into receive and send parts and provided by reconfigurable modules `NIC-RX` and `NIX-TX`. The block diagram of the network interface is shown in Figure 3.7. The network interface consists of hardware and software parts, which are controlled by an embedded operating system. Incoming packets are classified in the packet classifier and either passed to the video hardware or the receiver buffer (RECV FIFO). The management packets such as ARP, ICMP, TCP packets are stored in the receive buffer and processed in by the software network driver. Outgoing packets have two different sources: Packets containing video data sent from video hardware and network management packets are sent from software through the sender buffer (SEND FIFO). Because there are two paths producing packets, a packet multiplexer was added. The packets are multiplexed in a round-robin fashion. The buffers are connected to the embedded processor through the processor local bus. The packet classifier is also connected to the processor, the connection is not shown in the Figure.

The packet classifier is based on a content addressable memory (CAM). It contains memory for four classification rules, this size proved to be sufficient for our purposes. Based on the classification, each packet is passed to the video hardware, receiver memory or both. The purpose of the packet classification is to limit the rate of incoming packets to the embedded processor by selecting the interesting packets from 10 Gb/s incoming stream. The embedded processor is running at 150 MHz and therefore it cannot process all the packets from the 10 Gb/s interface.

A Linux network interface is created through the Linux TUN/TAP kernel module, which provides packet reception and transmission for user space programs. The program controlling the network hardware is running as a daemon in the user space, and through TUN/TAP module provides a new network interface. This new interface behaves like an ordinary network interface such as `eth0`. Therefore, the all networking services are available through this interface. A user space implementation was preferred over implementing the equivalent functionality as a kernel module because it simplifies the development cycle. The packet classifier is configured by a software framework described in the following section.

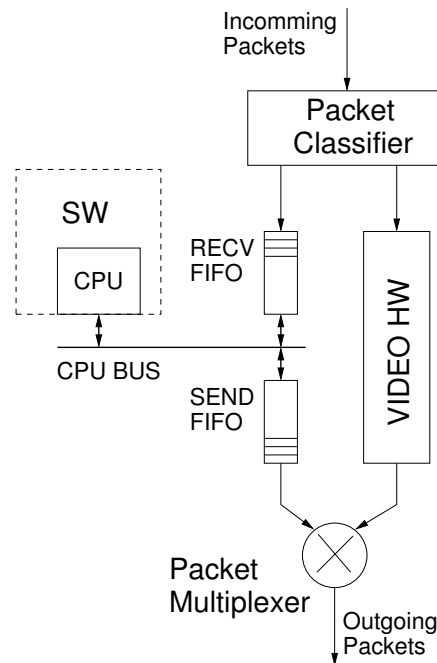


Figure 3.7: Ethernet Interface Block Diagram

3.2.2.5 Unified software framework for packet classification and filtering configuration

During the LOBSTER project [63] I proposed a generalized software framework for traffic filtering and classification configuration [A5] that enables transparent utilization of available hardware resources in different monitoring devices and which can be easily extended to support future types of hardware. I already extended the support to video and monitoring platforms.

The aim of the LOBSTER project was to develop and establish a pilot European infrastructure for accurate Internet traffic monitoring based on passive monitoring devices. The problem is that different monitoring devices have different resources for filtering and classification and are configured differently. This framework solves the problem.

This framework enables to specify the network classification rules in one common language and to produce specific configuration for several different types of monitoring hardware. The BPF (Berkeley Packet Filters) language is used because it is the most commonly used language in passive monitoring applications [64] that are our targeted use area.

The architecture is depicted in Figure 3.8. An application specifies classification rules in BPF language. The framework parses the rules into a device independent internal form (abstract syntax tree). The consistency of the internal form is checked and optimizations are performed (e.g. removing duplicate expressions). Then, the internal form is compiled and further optimised for a particular device-dependent backend (output generation). The framework supports output generators for the following devices: 1 Gb/s COMBO moni-

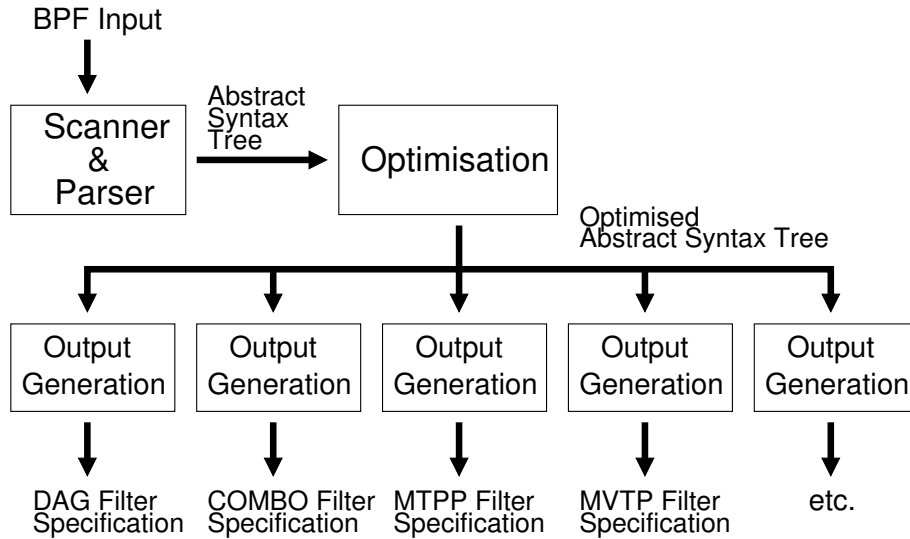


Figure 3.8: Architecture of transparent and extensible packet classification

toring cards [65], 10 Gb/s DAG monitoring cards [66] and video and monitoring platforms (MVTP and MTPP).

The proposed framework has the following features:

- Applications can run completely transparently in different hardware and software environments.
- The architecture is easily extendable with replaceable backends to support future hardware. Through the use of the intermediate abstract syntax tree which neither depends on the filtering specification language used nor on the hardware the filter will run, a clear separation between the two layers is achieved. One important advantage is that only the backend part generating the device-specific configuration needs to be rewritten when adding support for a new hardware platform.
- Filtering and classification specifications use abstract data structures applicable to current and future hardware.
- The classification rules are specified in the commonly known BPF (Berkeley Packet Filters) allowing easy porting of third-party applications using BPF filters.

The framework is published and presented in detail in [A5]. It was also incorporated into MAPI (Monitoring Application Programmable Interface) [64, 67] middleware authored by my colleague Vladimir Macek. MAPI is an open standard for development of passive monitoring applications. It has been developed as a part of the SCAMPI [68] and LOBSTER projects. MAPI allows to run multiple applications concurrently on multiple network cards where each application can monitor data from any of the cards in parallel with the other

applications. MAPI first tries to fit the monitoring requirements into the hardware. When hardware based classification cannot be used, a software implementation is used instead.

3.2.3 Evaluation

3.2.3.1 4K transmission in a loop over 14602 kilometers from Prague to Chicago

We tested our device over the GLIF (Global Lambda Interchange Facility) network in a loop Prague - Chicago - Prague in 2009. It was our first 4K (four HD channels) in-field test over a long distance commenced. The schematic diagram of the network setup is shown in Figure 3.9. The round-trip geodesic distance was approximately 9072 miles or 14602 kilometers. At the remote end in Chicago, a loopback was configured for this test. A video content in 4K resolution was used for this test, the 4K image was sent in four streams each in 2K resolution at 24 frames/sec. The average network throughput was 4.3 Gb/s. The 4K content was streamed from a Baselight Four color grading system located at the Cinepost corporation at Barrandov Studios in Prague.

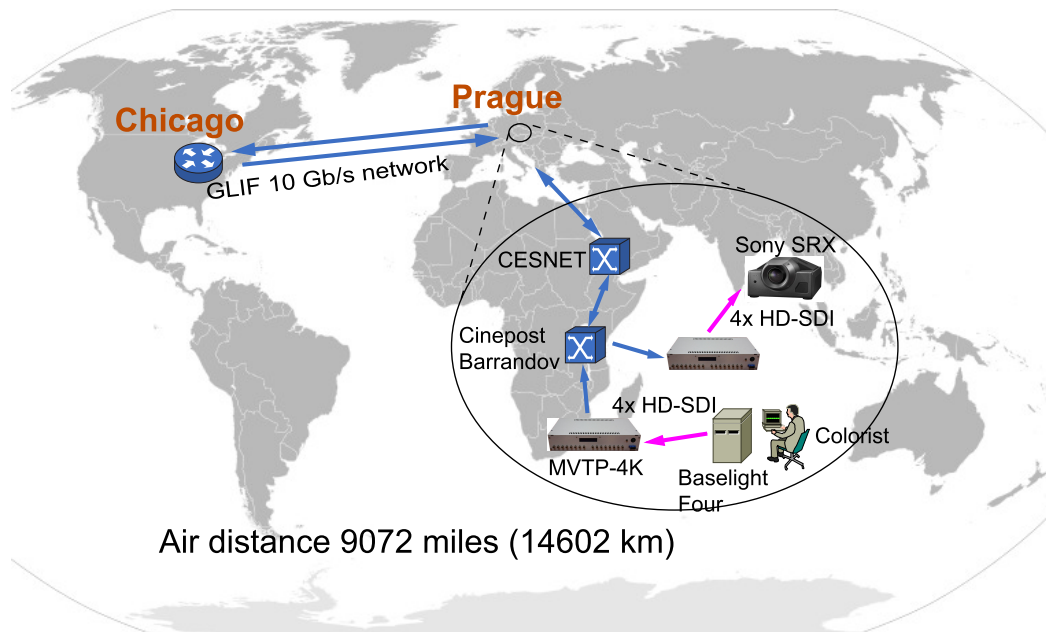


Figure 3.9: Remote loopback test (Prague-Chicago-Prague loop, 14602 kilometers)

In this test we verified the usability of the MVTP for the remote coloring demo for the CineGrid workshop 2009 [53] to be held 1 month later at the University of California, San Diego. The connection over the GLIF network consisted of a series of 10 Gb/s circuits inter-connected by an L3 router in Chicago and several L2 switches along the route. The used VLAN was not completely dedicated for our test and there was a small volume of other background traffic. The measured network latency was 200 msec.

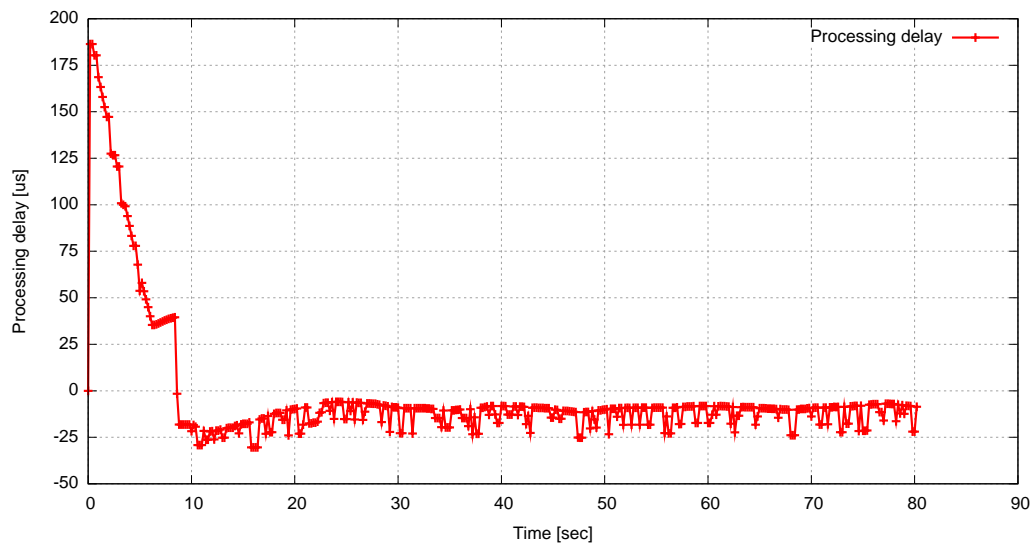


Figure 3.10: The processing delay measured in Prague-Chicago-Prague loop. Negative values mean that the lines arrived before they should be sent to the output.

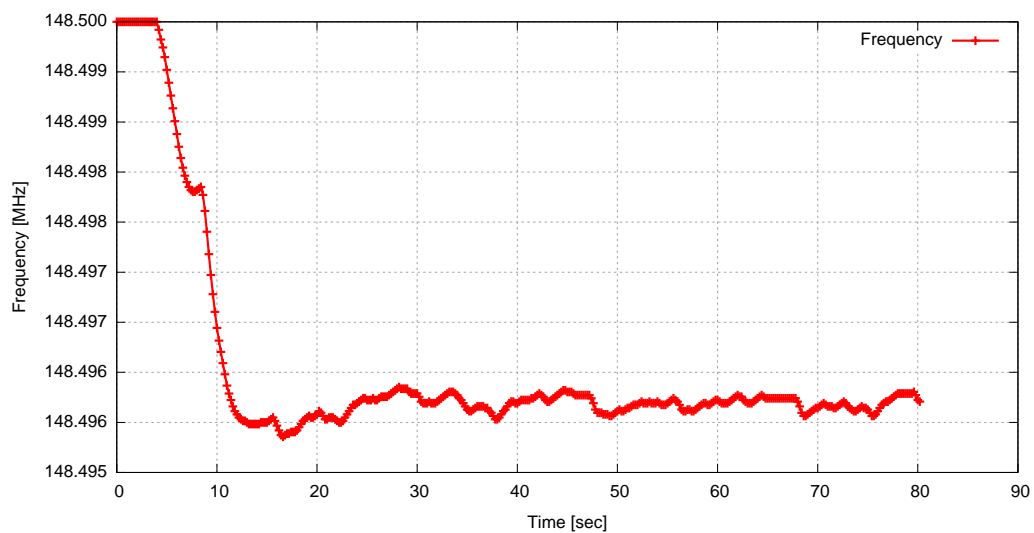


Figure 3.11: Regulation system response in Prague-Chicago-Prague loop. During the first 10 seconds the system is adapting to the sender's clock frequency.

The buffer capacity of the prototype was able to store 8 lines (0.3 msec) of 2K video image. In this configuration, network jitter caused occasional buffer overflow and thus a line drop at the rate of 3 lines per frame on average. We set the requested processing delay (requested buffer filling) to 12 μ s, close to the one half of a visible part of one 2K line. With this settings, the buffer was filled with a half of one line on average. The remaining buffer space was used for network jitter compensation. This settings was applied because we observed that the image lines (packets) were arriving in the bursts of packets followed by a delay. Setting the requested buffer filling to 50 % (as one can intuitively think that it is the best value) would only decrease the buffer space available for compensation of the packet bursts and increase the number of the lost lines.

Figure 3.10 shows the processing delay for the first 80 seconds after the transmission started. Positive values mean that the lines arrived after their expected rendering time causing the image to be shifted upwards. Negative values mean that the lines arrived before their rendering time and thus, are filling the buffer and increasing the network jitter resilience. After 10 seconds the processing delay settled around the requested (-) 12 μ s.

During the first 10 seconds one can see that the system is adapting to the sender's clock frequency and quickly tries to find the correct value. The regulator controls the tunable clock oscillator with center frequency 148.500 MHz. This frequency is divided by two inside the FPGA and is used for the all HD-SDI related logic. The regulation system response shows Figure 3.11.

Despite the network jitter, the proposed receiver synchronization was able to provide synchronous and stable video output with correct frame alignment. The excessive jitter was compensated by duplicating or removing single rows in case of buffer underflow or overflow. As a result, there were no subjectively observable image degradation. Later, we discovered that the excessive network jitter was caused by the loopback configuration in the L3 router in Chicago. The loopback configuration was removed for the CineGrid workshop and network jitter dropped significantly to approximately one line drop per several seconds.

3.2.3.2 HD transmission in loop over 35200 kilometers from Prague to Japan

Our next test was a loopback long-distance HD (720p) transmission from Masaryk hospital in Usti nad Labem in Czech Republic to the KEK research center in Tsukuba in Japan. The test was performed in 2010. The schematic diagram of the remote loopback network setup is shown in Figure 3.12. The round-trip geodesic distance was approximately 35200 kilometers.

In this test we verified the usability of the MVTP for the remote medial E-learning session between the Masaryk hospital and KEK research center to be held 3 month later. The network connection was formed by the CESNET network (from Usti nad Labem to Prague), the Pan-European Geant network (from Prague to Frankfurt), the link to the MAN LAN interconnection center in New York, the link to Japan and finally the SINET3 network in Japan where was the router with loopback configuration.

A video content was sent in HD (720p) resolution at 50 frames/sec. The average network throughput was 0.94 Gb/s, the measured network latency was 300 msec. The

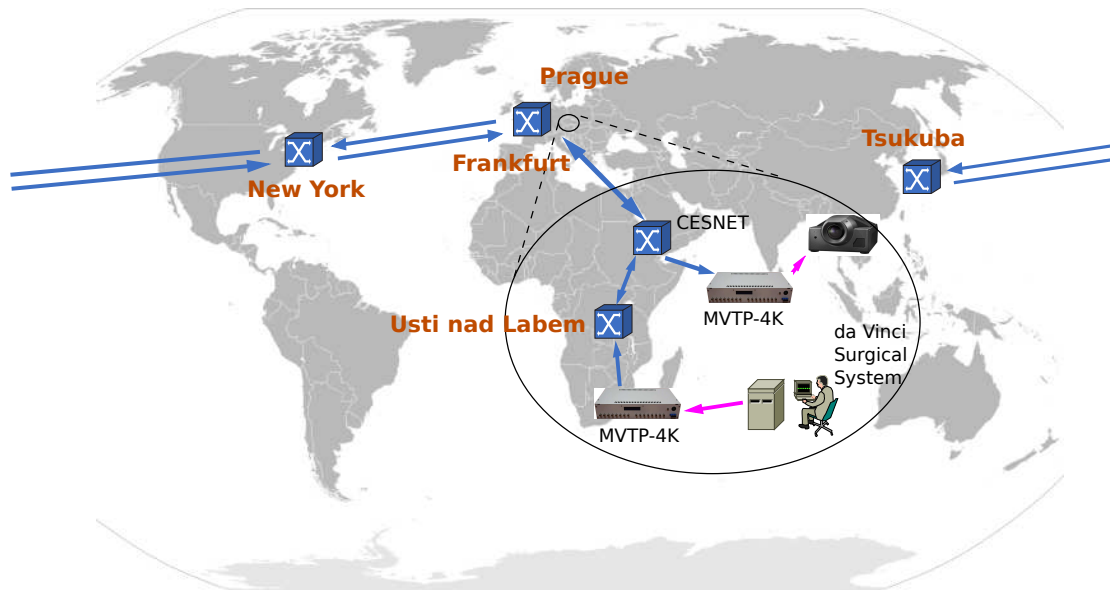


Figure 3.12: Remote loopback test (Prague-KEK in Japan-Prague loop, 35200 kilometers)

installed capacity of the all links was 10 Gb/s. However, we had to share the capacity of some of the links with other traffic amounting to 3-5 Gb/s.

When the HD resolution 720p at 50 frames/sec is used, the video packets are sent to the network with the following distribution (one line is sent in one jumbo packet):

- 99.86 % of packets (719 out of 720) with delta time 26.7 us – The time required for sending one line including the horizontal refresh (blanking) interval.
- 0.14 % of packets (1 out of 720) with delta time 826.7 us – The time includes the vertical refresh (blanking) interval of 30 lines.

Figure 3.13 shows the inter-packet arrival (delta) times for approximately 3 and a half minutes of traffic received from the loop.

The measured distribution of the delta times is as follows:

- 0.03 % of packets have delta time greater than 270 us (10 lines)
- 0.001 % of packets have delta time greater than 540 us (20 lines)
- 0.00016 % (12 packets) have delta time greater than 810 us (30 lines)

We increased the prototype receiving buffer capacity to store up to 30 lines (0.8 msec) of 720p video image. In this configuration, network jitter caused line drop at the average rate of 1 line per 30 seconds. Line drop at this rate are subjectively unnoticeable by humans.

The maximal total processing delay of the proposed system was 0.853 msec, i.e., 0.8 msec for receiver buffer and two times 26.7 us for input and output processing.

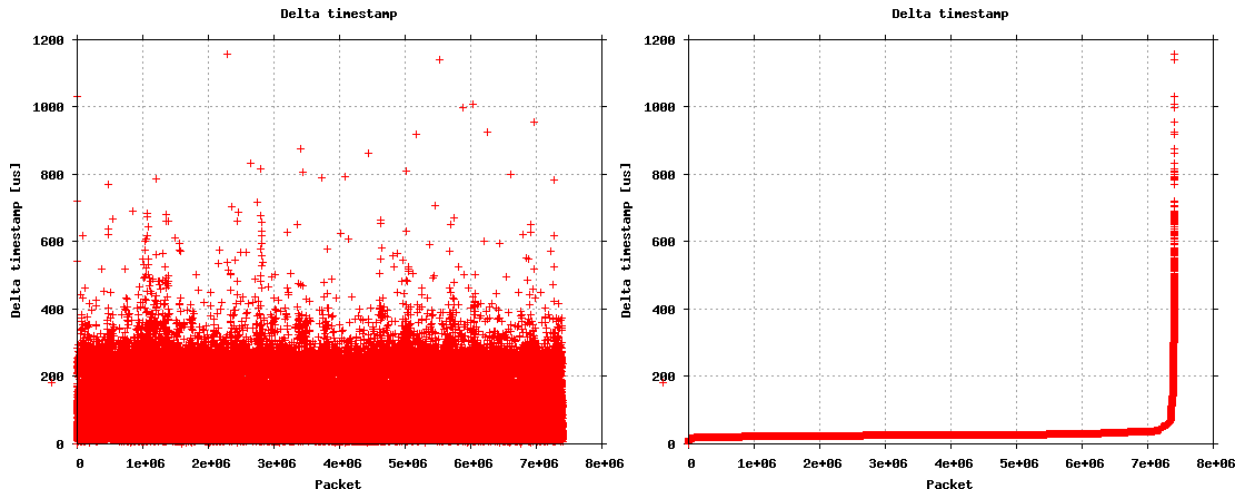


Figure 3.13: Inter-packet arrival (delta) times. The left plot shows the raw delta time measurement, the right plot shows the sorted values.

3.2.4 Summary

We have demonstrated that it is possible to transfer uncompressed high-resolution video signals over a long-distance asynchronous network with low-latency using FPGA technology. The proposed technology includes the video data serialization, network encapsulation and decapsulation as well as integration of replaceable modules for video data processing in one framework.

The receiver synchronization technique based on the processing delay described in the first contribution is the key part of this platform. We verified that the synchronization technique is able to synchronize the receiver to the sender with only small amount of line buffering.

The modular video transfer platform is characterized by:

- Real-time transfer of video content over a long-distance (> 10000 km) asynchronous network with no observable image degradation.
- Video input and output in multiple channel configurations (SDI, HD-SDI or 3G).
- Modular extendable design simplifying the addition of processing cores such as image transformation, compression or decompression.
- 10 Gb/s Ethernet network interface easily scalable up to 40 Gb/s and 100 Gb/s.
- Very small added latency, less than one msec.

After verifying the capabilities of the video processing platform, we demonstrated the real-time low-latency transmission over more than 10000 km from Prague to the University of California, San Diego in public where the Fourth Annual CineGrid workshop was

held [53]. The MVTP-4K was most likely the first device capable of low-latency high-definition bi-directional video transport of eight HD-SDI streams at that time.

The modular video transfer platform is published and presented in detail in [A3], which is included in Chapter 5. The platform is protected by Czech utility patent [A27] and is also partly covered by patents listed in the first contribution. We designed the two platform prototypes [A30, A29]. The first prototype is shown in Appendix A. This work was adopted by the Czech Technical Agency in project POVROS, whose goal is to move the results to the market. We got a project financed to transforming the prototype into a commercial product. The platforms were turned into commercially available products known as 4KGateway [11].

3.3 Prototype evaluation in a collaborative working environment

The last but not least contribution of this work is the application of the research results presented in the previous sections to practical applications. In this section we present an evaluation of the low-latency video transmission prototype MVTP-4K in the collaborative working environment. We present real-world use cases of remote real-time collaborations where the contributions described in this doctoral thesis have a verified impact on the productivity or allowed to work in ways which were unthinkable before. The presented use cases involve enhancements in several applications in the film industry, e-Learning in medicine, art and culture. Several use cases were conducted over a distance of more than 10000 km, across continents.

All the use cases were evaluated and are described along with observations and practical experience with emphasis on the contributions to the collaborative working environment.

3.3.1 Remote collaboration in multimedia industry

The use of MVTP-4K for remote collaboration in multimedia industry was demonstrated between continents over a distance of more than 6200 miles (10000 km) from Prague to the University of California in San Diego at several CineGrid workshops [53]. CineGrid is a non-profit membership organization whose aim is to build a multidisciplinary community that promotes research, development and adoption of technologies for the exchange of high-quality digital media over high-speed networks.

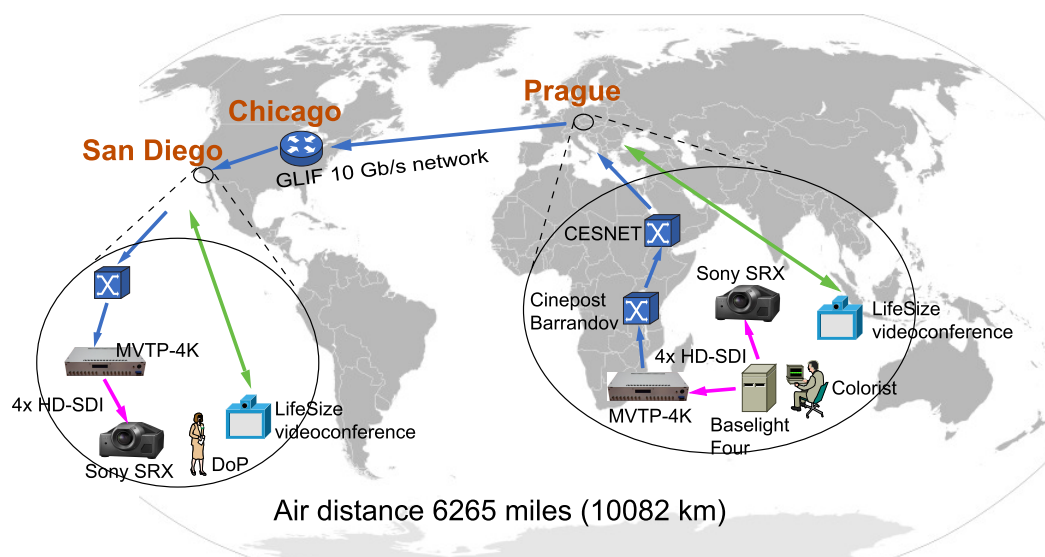


Figure 3.14: Schematic diagram of the network connection used during the remote color grading demonstration

At the CineGrid 2009 workshop we demonstrated the use of the described technology for real-time remote color grading of uncompressed 4K video, where the grading system and its operator (the colorist) were in the Barrandov Studios in Prague, while the Director of Photography, who instructed the colorist what to do and checked the results, was in San Diego.

The demonstration network setup is illustrated in Figure 3.14. The 4K content was streamed from the Baselight Four color grading system at the Cinepost corporation at Barrandov Studios in Prague. This content was transferred using two MVTP-4K devices over the GLIF network from Prague over Chicago to the University of California, San Diego (UCSD), where the CineGrid workshop took place. Additionally, there was also a bidirectional LifeSize video-conference connection between Cinepost and the CineGrid workshop venue. The network latency was 100 msec. The director of photography at the CineGrid workshop used this video conference link to discuss the color grading of the 4K content with the colorist at Cinepost, who performed the requested corrections in real-time on the 4K content streamed to the CineGrid workshop venue.



Figure 3.15: The screen configuration in lecture hall during the remote stereography demonstration

At the CineGrid 2010 workshop we successfully demonstrated the use of the technology described earlier for real-time remote stereography of stereoscopic (3D) 2K video. The processed content, the stereography software and the operator (the stereographer) were in

Prague at the Universal Production Partners (UPP), Inc., while the Director of Photography, who instructed the stereographer and immediately checked the results, was in San Diego.

The screen configuration in the lecture hall during the demonstration is illustrated in Figure 3.15. The network configuration was similar to the one used during the remote color grading demonstration but this time three different channels of 2K video were transmitted and simultaneously displayed. The projected image was displayed in 4K resolution using a Sony projector and the 4K image was divided into four independent quadrants of 2K resolution. The top left quadrant displayed the 2D image transmitted from Prague, the top right quadrant displayed the view of the lecture hall and finally the bottom left quadrant displayed a video stream from a camera in the studio in Prague. The bottom right quadrant was not used. The two channels of stereoscopic 2K image were displayed with two perfectly aligned JVC projectors on a silver screen. Each projector displayed the image for one eye using polarized light. The people in audience were wearing polarized glasses separating the images for the left and the right eyes. The low-latency transmission allowed to follow the work of the stereographer in real-time.

Summary

It was clear from the feedback of post-production experts that uncompressed transmissions preserving detail and color accuracy was highly acclaimed. This was only made possible through the use of the proposed architecture of high-volume real-time data processing.

The users said that despite the fact that latency due to network propagation delay was noticeable, they felt that the collaboration over the video link was genuinely interactive thanks to the minimal additional processing delay.

The experiments conducted over very long distances have shown that when the low processing delay on the sender and receiver side is added to the inevitable network propagation delay, the resulting response time is still well acceptable and gives the participants a feeling of interactivity. The team can work and see the image across 10000 km in real-time without any significant losses. This would save the persons involved a lot of traveling time and avoid problems associated to long distance traveling such as jet lag, speed up the workflow and therefore save money.

We published and presented the results of these tests in [A2], which is included in chapter 5 and in [A7, A10].

3.3.2 Remote collaboration and enhancements in e-Learning in medicine

The use of MVTP-4K for eLearning in medicine was demonstrated on multiple events including live stereoscopic (3D) streaming of robotically-assisted surgery from the da Vinci Surgical System [12] at the Masaryk Hospital in Usti nad Labem (Czech Republic).

The da Vinci surgical system is shown in Figure 3.16. The operator console (controlled by a surgeon) and the robot are usually located in the same room and connected by a

cable. The da Vinci robot is equipped with a stereoscopic camera which provides the surgeon with a 3D view of the surgical instruments and the tissue to be operated. A copy of the signal from this camera can be used for E-learning applications, such as remote medical training or presentations of surgical procedures on symposia. Also for complicated surgery procedures, advice of a remote expert during the surgery could be very helpful and would be made possible by a high resolution, low latency video transmission system such as the one presented here.

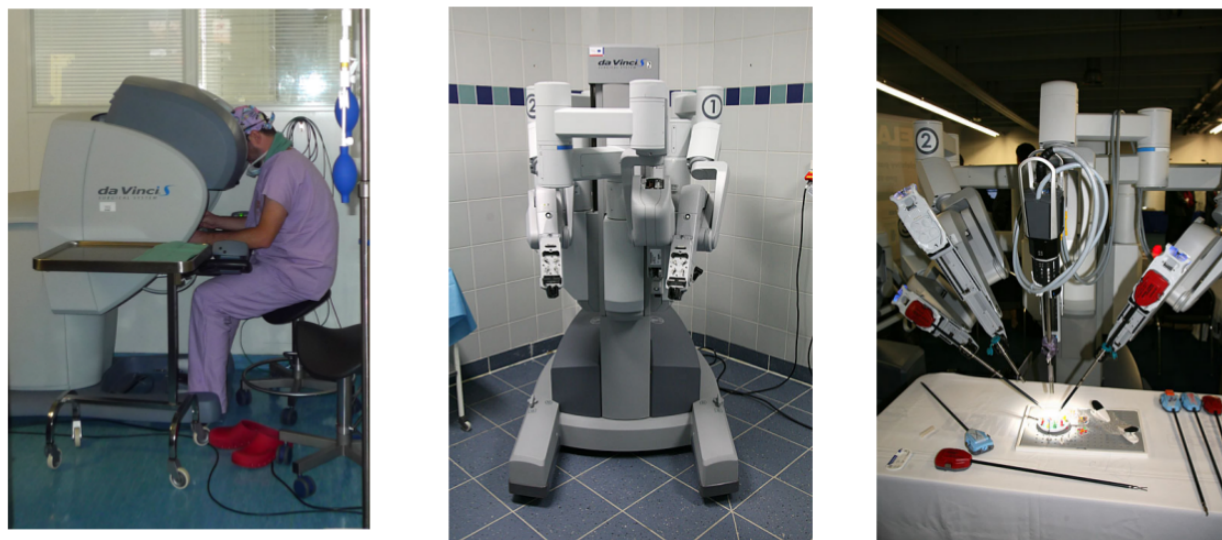


Figure 3.16: The da Vinci Surgical System, from left to right: The operator console, the operating robot, a detailed view of the robot arms with the surgical instruments (third instrument from the left is the stereoscopic camera).

The first demonstration of the system suitability for medical E-learning applications was a real-time transmission of two surgical operations from the Masaryk hospital to the 5th International Congress of Mini-invasive and Robotic Surgery in Brno (Czech Republic) in October 2010. The surgery was performed by MUDr. Jan Schraml, the head of the Department of Urology. A view of the audience room is shown in Figure 3.17. The surgeon commented the surgery and the medical experts in the audience asked questions and communicated with the surgeon in real-time.

Our second demonstration of the use for remote medical E-learning was a long-distance transmission from the Masaryk hospital to the KEK research center in Tsukuba in Japan, where a meeting with medical experts from Tsukuba University Hospital was held. The surgery was again performed and commented by Dr. Schraml. A parallel HD videoconferencing system was used by the audience at KEK to ask questions. The distance between the operation room and the audience was approximately 17600 kilometers. The networking delay was approximately 150 ms.

We tested the low latency video transmission platform also on another occasion between



Figure 3.17: Remote audience viewing and discussing the surgery during the 5th International Congress of Mini-invasive and Robotic Surgery in Brno (Czech Republic).

Europe and Asia in February 2011 during a live transmission to the Asia-Pacific (APAN) meeting in Hong Kong [69]. The distance was about 15000 km.

Summary

The feedback from participants in the long-distance medical events was very positive. Medical experts found it very useful and educative to see the operations without the need to travel long distances. They particularly appreciated the collaborative nature of long-distance discussions with the surgeon which was made possible by the low-latency transmission system. People asked impromptu questions, which were immediately answered by the surgeon. The stereoscopic projection also provided an immersive feeling and increased the educative value the seen content. The low latency of the transmission system allowed for a collaborative feeling.

We published and presented the details and feedback from the different transmissions in [A4], which is included in chapter 5 and in [A8, A9, A11]. The following CESNET press releases regarding the medical transmissions were issued: [70, 71]. I presented the MVTP-4K transmission with prerecorded 3D surgical demo in person at the ITU Telecom World conference in Geneva in October 2011 [72].

3.3.3 Remote collaboration in art, culture, science and engineering

The use-cases described here involve a remote collaboration in science and engineering, and remote access to cultural resources.

In cooperation with the Institute of Intermedia (IMM) [73] at the Czech Technical University in Prague we conducted several demonstrations of interactive high-resolution stereoscopic (3D) transmissions from a remote CAVE (Cave Automatic Virtual Environment) [8] – a device for immersive visualisation of spatial models. The CAVE system consists of a cube formed from five or six walls, a series of projectors projecting a stereoscopic image on each wall, a rendering computer cluster rendering a stereoscopic image of a 3D model for each of these walls and a device measuring the position of the viewer inside the cube.

At the Terena Networking Conference 2011 in Prague we demonstrated an interactive access to a 3D model. The 3D model was rendered at the IMM, displayed in the local CAVE system and the high-resolution stereoscopic image of one wall was transferred and displayed at the conference venue. The link was bidirectional, people visiting the conference were able to control the display and move through the displayed model using an attached keypad device. The commands were transferred back to the rendering device at the IMM.

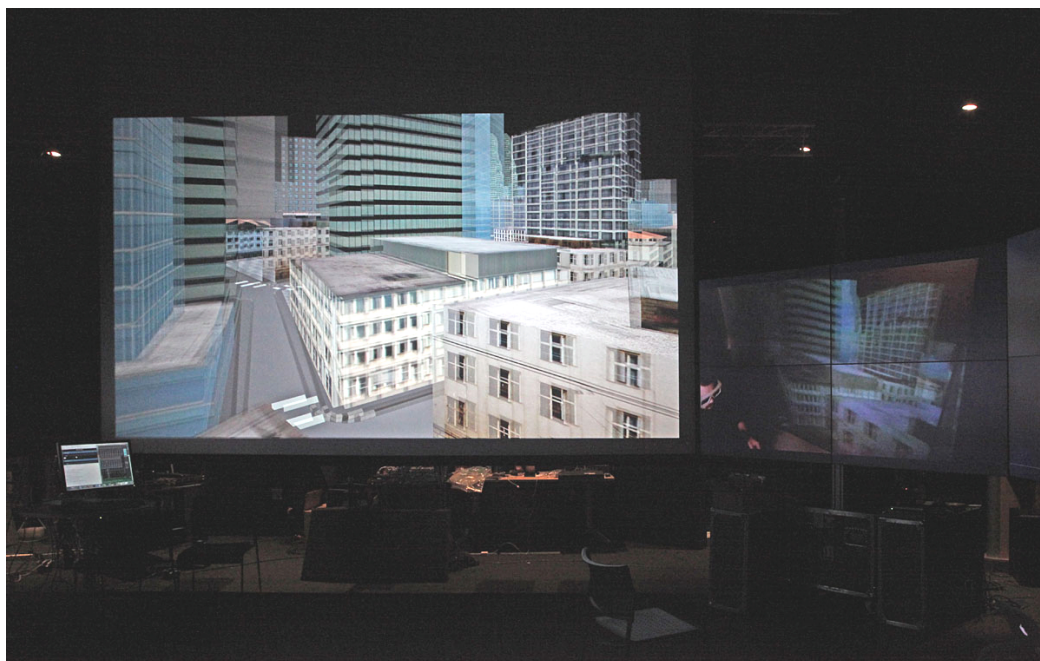


Figure 3.18: A view of the remote audience room during the CineGrid 2011 presentation. The image looks blurred because the images for both eyes appear superimposed.

During the CineGrid 2011 workshop in San Diego we demonstrated showed two different applications. The first application demonstrated how remote interactive access to a 3D

architectural model can enable collaborative design of a new building complex. The setup was similar to the one presented at the Terena Networking Conference with the exception that now the distance between the participants was more than 10000 km. People from both sides exchanged ideas in real time on potential improvements of the buildings to be constructed. A view of the audience room is shown in Figure 3.18. The main screen showed the 3D projection of the architectural model to be worked on. A smaller side screen showed a view of the remote CAVE, to keep better social contact with remote persons.

The third application demonstrated a remote interactive access to a 3D model, which cannot be moved otherwise. It was a computer model of the Langweil model of Prague 2, which is a physical paper model of the historical center of Prague. It was created in 1826-37 by Antonin Langweil in 1:480 scale and includes approx. 2000 houses. The model is currently on display at the City of Prague Museum. It has a significant historical value because approximately half of the buildings no longer exist. Due to its size and fragile conditions the physical model cannot be moved. The intellectual property rights to the digitized model also do not permit sharing them outside the museum organization. It is however allowed to share a visualization of selected parts of the model. Rendered images of the model were streamed over a distance of more than 10000 km. People at the remote conference site could navigate through the streets of the model interactively and study details of individual historical buildings.

Summary

The feedback from participants in the long-distance remote interactive visualization was again very positive. People said that the delay between the movement of the control device and a corresponding change in visualization is noticeable (approximately 250 msec latency), but it can be well accommodated in thinking and interactive feeling is still possible. This can be considered a satisfactory achievement in given large distance between the remote sites.

We showed that it is possible to transmit the stereoscopic images rendered from a system which cannot be moved otherwise. The rendering cluster together with the 3D models are located in one place, the rendered image is transferred across the network and displayed and controlled in different locations, e.g. in a remote CAVE system. When combined with our low-latency video transmission system, the interactive feeling is still preserved. As a result, traveling time and money are saved.

The system for experimental remote access to 3D models was developed by my colleague Zdenek Travnicek. It consists of a hybrid architecture - a software video sender, hardware receiver and control system based on a keypad device. The software sender is directly running on the CAVE cluster. The MVTP-4K is used as a hardware video receiver. The keypad device is attached to a laptop running a software sending the commands back to the CAVE cluster.

We published and presented the details and feedback from the transmissions in [A6]. CESNET also issued a press release [74] on the demonstration of remote collaboration at the CineGrid 2011 workshop.

Chapter 4

Conclusions

4.1 Summary

In this doctoral thesis I address the problems of low-latency video transmissions with the emphasis on the contributions to the collaborative working environment.

In the first contribution, I proposed a new and now patented technique based on an asynchronous clock recovery for the receiver to sender video synchronization. This technique makes use of small memory buffers (much less than the buffer required for one image frame). This allows direct implementation in an FPGA without the need for the external memories, thus reducing costs for the device and adding a minimal processing delay less than 1 msec. This is much smaller than with previous devices. Due to the small processing delay, this technique allows to conduct transmissions with very low latency with minimal hardware requirements over asynchronous networks such as Ethernet or more generally Internet.

The second contributions describe the proof-of-concept architecture of a prototype developed during the research. The proposed synchronization technique is a key part of the prototype. The prototype is capable of low-latency video transmissions with resolutions up to 4K in stereoscopic mode (3D). The prototype is based on reconfigurable plug-in modules. I proposed an architecture simplifying the module preparation and reducing the complexity of the hardware required for the module reconfiguration. I also developed a packet filtering and classification module and proposed a new extendable software framework for the configuration of packet filtering and classification. The framework allows to configure various types of standalone hardware and PC network monitoring cards with classification rules specified in only one language.

The last but not least contribution of this doctoral thesis is an evaluation of the proposed architecture in a collaborative working environment, where the low latency is important and improves the collaborative feeling and working efficiency. We conducted real-world demonstrations of real-time remote collaborations with focus on applications in the film industry, e-Learning in medicine, art and culture, where the contributions described in this doctoral thesis had a verified impact on the productivity and enabled completely new

ways of working. In particular, they improve the quality of teaching and training, allow an interactive participation in remote or normally inaccessible places, save a lot of traveling time, be more environmentally friendly, speed up the whole workflows and therefore save money. Several demonstrations were conducted over a distance of more than 10000 km, across continents. The observations along with practical experience were also presented.

The low-latency video transmission prototype was adopted by:

- Film & TV post-production company Cinepost at Barrandov Studios in Prague for daily transmissions between several remote studios.
- Institute of Health Studies of Jan Evangelista Purkyně University in Ústí nad Labem to enhance e-learning in medicine.
- Czech Technical Agency in project POVROS. We got a project financed to transforming the prototype into a commercial product. The prototype was turned into a commercially available product known as 4KGateway [11].

4.2 Future Work

The technology described in this doctoral thesis enables distributed teams to significantly increase their productivity by sharing video content in real time. In our future work we plan to continue with the research in the field of collaborative working environments and we suggest to explore the following:

- Investigate which video processing functions would be useful to incorporate in the hardware, such as encryption, transcoding, image manipulation or different types of compression. There is however a trade-off between the requested functionality and added processing latency. An 8K uncompressed version over a 40 Gb/s network is technically possible if there is demand.
- Investigate an automatic adaptation of multimedia content according to capabilities of the network environment and end-point visualization devices. This should include adaptation to the available bandwidth (possibly by dynamic changes of compression levels), number of information channels (e.g., 2D or 3D) or security requirements by encrypted streaming.
- Investigate inter-stream synchronization which keeps the stable latency not only between the two end-points but also between multiple end-points. This should be possible by researching a distributed variant of reactive play-out buffer management discussed in Section 2.1.5. This will guarantee a synchronous communication and consistent playout across several separated locations.
- Find new applications and new ways of working and promote them by realizing real use-case demos and field trials. This also provides important feedback from the public

audience to the researchers and can help to identify the missing parts and set the correct direction for the future developments.

Chapter 5

Publications Included in Thesis

This chapter contains the key publications included in this doctoral thesis. Every section of this chapter presents one publications and starts with related information about the publication.

Several publications include a grayed text that signify that the text belongs to other authors and is not considered as a part of the research results described in this doctoral thesis. These publications are explicitly marked.

5.1 J. Halak, S. Ubik, and P. Zejdl. Receiver synchronization in video streaming with short latency over asynchronous networks

This paper was published in: *Proceedings of the 13th IEEE Symposium on Design and Diagnostics of Electronic Circuits and Systems* in 2010 [A1].

Receiver synchronization in video streaming with short latency over asynchronous networks

Jiří Halák, Sven Ubik, Petr Žejdl
CESNET
Email: {halak,ubik,zejdl}@cesnet.cz

Abstract—Video transfers are an expected driver application area of the future Internet. When sending real-time video signals over an asynchronous network, such as Ethernet, some mechanisms need to be used to adjust the rendering speed on the receiver to the data source rate on the sender. The problem is more difficult when an ultra high-definition resolution is used (such as 4K x 2K). In this paper we discuss design options and our practical experience with implementation and field trial.

I. INTRODUCTION

Video transfers are an expected driver application area of the future Internet. Better-than-high-definition resolution, such as 4K x 2K (4096x2160) is already required in some application areas, such as scientific visualisation and post-production in film industry. Uncompressed video streams are preferred when ultimate quality and low latency for remote collaboration is required. Compression algorithms increase latency and may reduce picture quality.

The real-time video streaming requires that the speed of rendering on the receiver side matches the rate of video source on the sender side. When the sender and receiver are connected over an asynchronous network, such as Ethernet, the receiver cannot directly synchronize its clock with the sender. The problem is more difficult when audio channels need to be transferred along with video and when an ultra high-definition resolution is used (such as 4K x 2K), which usually requires sending a picture in multiple synchronized parts of the screen (four quadrants in case of 4K x 2K).

Therefore, the receiver needs to implement a technique that adjusts to the speed of data arrival and still maintains a continuous stream of video and audio data to the rendering device, preferably with minimal added latency for remote interactive applications.

In this paper we discuss design options to resolve this problem for transmissions of ultra high-definition resolution (4K) video streams with embedded audio. We also describe our practical experience with implementation and field trial.

II. REQUIREMENTS

We set the following set of requirements to be fulfilled by the proposed solution:

- video input and output in 4x dual-link HD-SDI channels
- 10 Gigabit Ethernet network interface
- no visual impairments due to clock difference or network jitter
- pixel correct synchronization between picture quadrants

- audio synchronized with video
- small added latency
- fit into available FPGA circuits

The use of a dual-link HD-SDI channel for transmission of high definition video streams is now a common industry practice and it is specified in SMPTE 372 [1]. This includes HD (1920x1080) and 2K (2048x1080) formats. The 4K (4096x2160) signals are typically transferred in four quadrants, each in 2K format carried over a separate dual-link HD-SDI channel.

The FPGA circuit was chosen as the processing device due to its versatility that allows to combine video transmissions with other functions, such as compression, encryption or transcoding.

III. DESIGN DECISIONS

There are two sources of rate difference between the data arriving to the receiver from the network and the data to be sent to the rendering device. First, the internal clock of the sender can be different from the internal clock of the receiver, within a tolerance permitted by the respective transmission protocol formats. The HD-SDI clock rate is specified as 1.485 Gb/s or 1.485/1.001 Gb/s with 100 ppm tolerance. Second, jitter can be introduced due to network traffic conditions. The network jitter needs to be accommodated by the receiver FIFO memory.

A. Adjusting the rendering rate

Several alternative techniques can be used to adjust the sending and rendering rate over an asynchronous network: receiver feedback, frame buffer, blank period adjustments and rendering clock adjustments.

a) Receiver feedback: The receiver can send feedback to the sender requesting sending rate adjustments (flow control). This technique is used in window-based transport layer protocols, such as TCP or in some link layer protocols, such as PAUSE frames in Ethernet.

The adjusted data rate appears at the receiver after round-trip time (RTT), which is typically hundreds of milliseconds on the long-distance Internet connections. In high-definition video transfers, this technique would require a large receiver FIFO memory, which would not fit in FPGA resources and which would introduce high added latency. Therefore we decided to use another technique.

b) *Frame buffer*: This technique requires the receiver to have a FIFO memory large enough to accommodate at least three complete frames. Then the rendering device can be driven by a fixed clock source in the receiver. The top of the FIFO memory holds the frame that is being sent to the rendering device. When sending of this frame completes, the next FIFO position should include the next frame to be sent. The third (and possibly further) FIFO positions can hold more data depending on network jitter.

After a long period when the skew between the sender and receiver clock rates causes that the frame in the second FIFO position is not completely available in time, the frame in the first FIFO position is rendered again. Similarly, if the FIFO cannot accommodate more incoming data, one frame is dropped. Of course, this solution is not suitable when audio signals are transferred along with video signals.

In the worst case when both the sender and receiver clocks are shifted by 100 ppm in the opposite directions, the error can expand to the whole frame in $1/200 * 10^{-6} = 5000$ frames. At 24 frames per second, it equals to approx. 3.5 minutes. A large FIFO memory can extend this time.

Unfortunately, we could not use this solution due to limited memory resources in the FPGA circuit.

c) *Blank period adjustments*: Most digital video transfer formats include a blank period in addition to an active period (visible lines) in each frame. The blank period is used for embedded audio, ancillary data and synchronization. The structure of the 2K frame [2] (or one quadrant of the 4K frame) as transmitted using a dual-link HD-SDI channel is shown in Fig. 1. Each frame consists of 1125 lines, which include 1080 active lines. Each line consists of 2750 samples, which include 2048 active samples. The active period is shown in gray color. The SAV (start of active video) and EAV (end of active video) are synchronizing sequences present in each line.

One simple technique to adjust the rendering speed to the rate of incoming data is to add or remove some samples or lines in the blank period. While this technique does not comply with SMPTE standards, we tested that it works with some devices. Since we required compliance with standards, we also decided not to use this technique in our final solution.

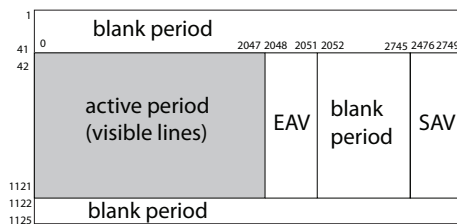


Fig. 1. Format of 2K frame

d) *Rendering clock adjustments*: Adjusting the clock in HD-SDI channels between the receiver and the rendering device within the permitted tolerance gives the receiver some level of adaptation to the rate of incoming data. This solution

requires a tunable oscillator and a closed-loop controller in the receiver. It is a solution that we implemented.

B. Controller

In order to adjust the rendering clock to the data source rate, we used a common PID controller [3]. The complete receiver control structure is shown in Fig. 2. The FIFO input clock is driven by data arriving from the network. The FIFO output clock is driven by a clock generator, which is tuned by the PID controller.

The bare adjusting of the rendering clock would stabilize the rendered picture, but it does not guarantee the frame alignment. The top of the picture could appear anywhere on the screen.

Therefore, we used as the feedback variable to drive the controller the delay between the time when the first active line enters the receiver FIFO memory and the time when the receiver starts to send the first active line to the rendering device measured in clock cycles.

This delay is written to a register for each frame and sampled by the regulator every 200 ms. The controller then uses a weighted moving average of eight last samples. The purpose of this average is to smooth out fluctuations in FIFO occupancy due to network jitter.

One may argue why we do not use the time when the first active line leaves the FIFO instead. This would actually introduce instability, which we confirmed in our tests. If the FIFO occupancy increases due to the receiver clock being slower than the sender clock, packet loss at the FIFO input will increase as a result of network jitter. This will increase probability that the first active line (or any other line used by the regulator) is also lost. And this will cause the regulator slower to react thereby allowing further increase in the FIFO occupancy and higher packet loss.

The feedback variable is compared with the desired value, which is the number of cycles it takes for data to pass through the receiver FIFO at a desired average occupation added to the latency of the output video processor. This value should be optimized empirically for maximum picture stability, see section IV. The PID controller then produces adjustments to the clock generator frequency.

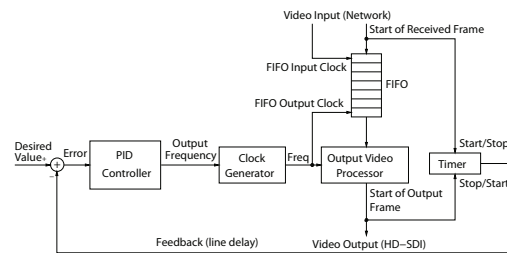


Fig. 2. Receiver control structure



Fig. 4. Remote loopback test (Prague-Chicago-Prague loop, 14602 kilometers, picture taken in Cinepost, see Acknowledgements)

C. Out of range adjustments

The receiver FIFO can accommodate some fluctuations in data arrivals on its input end. But when a line is not available at the output end when it should be rendered, the receiver sends another line to the rendering device, because the signal on the synchronous HD-SDI channels cannot be stopped. This is either the line, which happens to be at the FIFO output end or a copy of the previous line if FIFO is empty. Also, when the newly arriving line does not fit into the receiver FIFO, it is dropped. These events causes the whole picture to roll up or down. The controller will see large value of the error variable and try to compensate quickly. See also section IV for our practical observations of impairments.

IV. PRACTICAL EXPERIENCE

We implemented the described architecture on the Xilinx Virtex 5 FPGA device. The software part runs on an embedded Linux, which runs inside the FPGA using the softcore Microblaze [4] processor. The Linux environment provides access to firmware variables and runs the PID controller process, which communicates with a clock generator over an RS-232 interface.

Without a controller, the picture started to roll down or up after about 3-4 seconds even when communicating over a single 10 Gigabit Ethernet segment. This was due to the difference and wonder of the sender and receiver clocks. When the controller was activated, the picture was perfectly stable. The controller reads the feedback variable, computes the error variable and generates commands for the clock generator periodically. Most fluctuations in data arrivals were due to network jitter, which was handled by the FIFO memory.

A few lines of the PID controller output are shown in Fig. 3. Each lines indicates one 200 ms adjustment. The number show from left to right: feedback variable in clocks, moving average of feedback variable in clocks, change of this average between two samples, resulting frequency to be set on the clock generator in Hz and change of this frequency in Hz.

```
raster= 779 avgf= 1034 speed\_avgf= -45.20
freq=148496064 df= 21.26
raster= 1778 avgf= 1220 speed\_avgf= 12.63
freq=148496048 df= -9.51
raster= 750 avgf= 1102 speed\_avgf= -19.89
freq=148496048 df= 7.92
```

Fig. 3. Example of PID controller output

We tested our device over the GLIF (Global Lambda Interchange Facility) network in a loop Prague - Chicago - Prague. The air distance was approx. 9072 miles or 14602 kilometers. In this configuration, network jitter was much higher and frequently exceeded the FIFO capacity. By duplicating or removing single rows complemented with rendering clock adjustments by the controller, the impairments were not subjectively observable. An overall impression from this remote loopback test is shown in Fig. 4.

V. RELATED WORK

Net Insight's [5] Nimbra 600 series switch can transport 8x HD-SDI or 3G SDI channels over an SONET/SDH network. There are several commercially available solutions for transport of compressed 4K video over the Internet, for example NTT Electronics [6] ES8000/DS8000 4K MPEG-2 encoder/decoder complemented with NA5000 IP interface unit and intoPIX's [7] system of PRISTINE PCI-E FPGA boards and JPEG 2000 IP cores.

The system described in this article differs in that it transports uncompressed 4K video streams for ultimate quality over an asynchronous Ethernet network.

VI. CONCLUSION AND FUTURE DIRECTIONS

We have demonstrated that it is possible to transfer an uncompressed 4K video signal over a long-distance asynchronous network in real time without observable visual disturbances using FPGA technology for data framing and deframing and common control elements (PID controller). We used a novel method of stabilizing the picture on the receiver side without a synchronous network for clock recovery.

In our further work we will investigate options to better support applications, such as bidirectional transfers and various synchronization requirements among channels. Uncompressed 8K video over a 40 Gb/s network is feasible if there is demand.

REFERENCES

- [1] *Dual Link 1.5 Gb/s Digital Interface for 1920 x 1080 and 2048 x 1080 Picture Formats*, SMPTE 372-2009, Society of Motion Picture and Television Engineers.
- [2] *D-Cinema Distribution Master - Image Pixel Structure Level 3 - Serial Digital Interface Signal Formatting*, SMPTE 428-9-2008, Society of Motion Picture and Television Engineers.
- [3] Bela Liptak. *Instrument Engineers' Handbook, Third Edition: Process Control*, Butterworth-Heinemann, ISBN 0801982421.
- [4] MicroBlaze Soft Processor Core, <http://www.xilinx.com/tools/microblaze.htm>.
- [5] Net Insight AB, <http://www.netinsight.se>.
- [6] NTT Electronics, <http://www.ntt-electronics.com>.
- [7] intoPIX, <http://www.intopix.com>.

A. Acknowledgements

This work has been supported by the Ministry of Education, Youth and Sports of the Czech Republic under the research intent MSM6383917201 "Optical Network of National Research and Its New Applications".

The tests using the Baselight Four and the Sony SRX 4K projector were possible thanks to the Cinepost corporation.

5.2 J. Halak, M. Krsek, S. Ubik, P. Zejdl, and F. Nevrela. Real-time long-distance transfer of uncompressed 4K video for remote collaboration

This paper was published in: *Future Generation Computer Systems Journal* in 2011 [A2].

This paper includes a grayed text that is not considered as a part of this doctoral thesis. The grayed text describes the research results proposed by paper co-author Jiří Halák and are considered as a part of his dissertation thesis. These results were only used and are not proposed by me.



Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Real-time long-distance transfer of uncompressed 4K video for remote collaboration

Jiří Halák^{a,*}, Michal Krsek^a, Sven Ubik^a, Petr Žejdl^a, Felix Nevřela^b

^a CESNET, Žitkova 4, Prague 6, Czech Republic

^b CINEPOST, Kříženeckého Nám, 5/322, Prague 5, Czech Republic

ARTICLE INFO

Article history:

Received 26 February 2010

Received in revised form

16 June 2010

Accepted 12 November 2010

Available online 5 December 2010

Keywords:

Video

Color

Collaborative computing

Remote systems

Network communication

High-speed

ABSTRACT

Better-than-high-definition-resolution video content (such as 4K) is already being used in some areas, such as scientific visualization and film post-production. Effective collaboration in these areas requires real-time transfers of such video content. Two of the main technical issues are high-data volume and time synchronization when transferring over an asynchronous network such as the current Internet.

In this article, we discuss design options for a real-time long-distance uncompressed 4K video transfer system. We present our practical experience with such transfers and show how they can be used to increase productivity in film post-production, as an application example.

© 2010 Elsevier B.V. All rights reserved.

1. Introduction

Video transfers are an expected driver application area of the future Internet. Picture resolution has been increasing over time. Better-than-high-definition-resolution video (such as 4K) is already used in some areas, such as scientific visualization and the film industry. For the ultimate quality, required for instance in the color-grading process in film post-production, working with a signal that has not been compressed is preferable.

Presentation of high resolution video is now well possible using rendering devices with the corresponding resolution, using tiled displays (e.g., SAGE [1,2]) or multi-dimensional systems (CAVE [3,4]). The productivity of a distributed team can be significantly increased when the video signal can be transferred over the network in real time, to discuss and perform the processing of video content. Two of the main technical issues are high-data volume and time synchronization when transferring over an asynchronous network such as the current Internet.

For 4K resolution, the data volume ranges from 4.2 Gb/s for 4:2:2 subsampling [5], 10-bit color depth and 24 frames per second

to over 9.6 Gb/s for RGB (no subsampling), 12-bit color depth and 30 frames per second. Overhead in packet headers needs to be added.

Real-time video streaming requires that the speed of rendering on the receiver side matches the rate of video source on the sender side. When the sender and receiver are connected over an asynchronous network, such as Ethernet, the receiver cannot directly synchronize its clock with the sender.

Therefore, the receiver needs to implement a technique that adjusts the speed of data arrival and maintains a continuous stream of video and audio data to the rendering device, preferably with minimal added latency for remote interactive applications.

We implemented the proposed architecture in a device called MVTP-4K (Modular Video Transfer Platform).

The structure of this paper is as follows. In Section 2, we summarize the main system requirements for real-time high-definition video transfers. The proposed system architecture is described in Section 3. Our practical experience is described in Section 4. Related work is referred to in Section 5 and our conclusions and thought about future directions are provided in Section 6.

2. Requirements and design constraints

In order to satisfy the needs of the targeted applications, we set the following set of requirements to be fulfilled by the proposed solution:

* Corresponding author.

E-mail addresses: halak@cesnet.cz (J. Halák), michalk@cesnet.cz (M. Krsek), ubik@cesnet.cz (S. Ubik), zejdlp@cesnet.cz (P. Žejdl), felix.nevrela@cinetpost.cz (F. Nevřela).



Fig. 1. Hardware architecture.

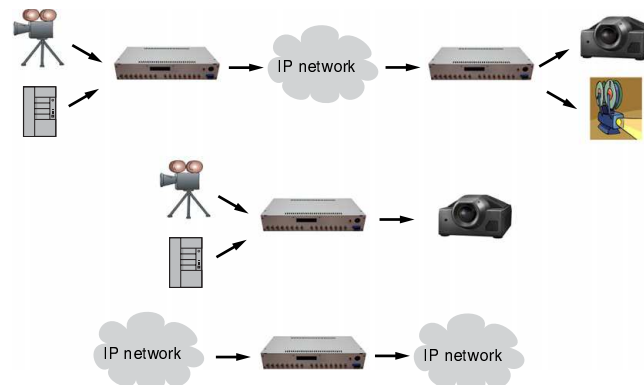


Fig. 2. Operation modes—network transfer (above), video processor (center), network processor (bottom).

- Real-time transfer of uncompressed 4K video content over a long-distance asynchronous network with no observable visual impairments.
- Support of at least 24 frames per second, 12-bit color depth and no subsampling (RGB).
- Video input and output in multiple HD-SDI channels.
- 10 Gb Ethernet network interface.
- Pixel-correct synchronization between picture quadrants.
- Audio synchronized with video.
- Small added latency (see below).

The rationale behind these requirements, in addition to those already mentioned in Section 1, is as follows.

The use of HD-SDI channels for transmission of high definition video is now a common industry practice. Three variants are currently in use – HD-SDI [7], dual-link HD-SDI [8] and 3G-SDI [9]. Mapping of digital video signals into these channels is specified for HD (1920 × 1080) [10,8], 2K (2048 × 1080, D-Cinema operational level 2 and 3) [8] and for other lower resolution formats. The 4K format (4096 × 2160, D-Cinema operational level 1 or 3840 × 2160, UHDTV1 [11]) is typically transferred in four quadrants, each in 2K or HD format carried over a separate HD-SDI channel.

Asynchronous Ethernet technology is currently more frequently deployed in 10 Gb/s networks than synchronous SONET/SDH, due to its simplicity and therefore lower cost. Ethernet will likely play even more important role in future 40 Gb/s and 100 Gb/s networks, although often enveloped in a synchronous Optical Transport Network (OTN) [12].

Empirical experience has shown that the maximum acceptable one-way latency for remote interactive work not perceived by users as a limiting factor is around 150 ms [6]. This latency can easily be caused just by network propagation delay. Therefore, the video transfer system should add minimal further latency. Buffering of one frame at 24 frames per second adds 41 ms.

3. Architecture

3.1. Hardware

Real-time processing of multi-gigabit data rates is difficult on PC-based platforms with standard operating systems not designed

for real-time operation. We were looking for a real-time design that is scalable to higher data rates (such as for 8K or UHDTV2 format), higher network speeds (such as 40 and 100 Gb/s) and that can be integrated with commonly requested video processing functions, such as encryption, transcoding or compression. This implies highly parallel and truly real-time data paths. DSP (digital signal processor) and FPGA (field programmable gate arrays) are the standard technologies in this area. We selected FPGA, due to its high data bandwidth and our design having no requirements with regard to floating-point operations.

The selected FPGA circuit needs to have a sufficient number of fast channels for input and output of the HD-SDI data. The sustained speed needs to be 1.485 Gb/s for HD-SDI and 2.97 Gb/s for 3G-SDI. For the 4K format, we need four or eight HD-SDI interfaces depending on the exact format and interface speed. Xilinx, Altera and Lattice all have FPGA circuits that satisfy these requirements. We selected Xilinx Virtex 5, which allowed us to reuse some design blocks that we have developed for network monitoring devices.

The hardware architecture is shown in Fig. 1. The HD-SDI board converts electrical levels and timing between input and output HD-SDI channels on the one side and Virtex RocketIO channels on the other side. The FPGA board processes the video signal and is connected to an optical transceiver, which converts electrical and optical signals for network transmission.

This architecture allows operation in several modes illustrated in Fig. 2. Two devices can be used to transfer video content over a network or a single device can be used as a video processor or a network processor.

3.2. Packetization

There are several possible ways of mapping the HD-SDI data into network packets. Three options are described below. The resulting bit rate for 4K (all four quadrants) at the physical layer is summarized in Table 1. These rates also include embedded audio and the packet header overhead. We assume 24 frames per second as per the D-Cinema format.

Complete HD-SDI data.

One solution is to transfer all HD-SDI data. One complete line of the 2K frame (one quadrant) consists of 2750 samples. One

Table 1
Bit rates at the physical layer.

	10-bit 4:2:2 (Gb/s)	10-bit RGB (Gb/s)	12-bit 4:2:2 (Gb/s)	12-bit RGB (Gb/s)
Complete HD-SDI data	6.00	12.00	12.00	12.00
Just image & audio bits	4.58	6.50	5.22	7.77
Active area samples	4.59	9.18	9.18	9.18

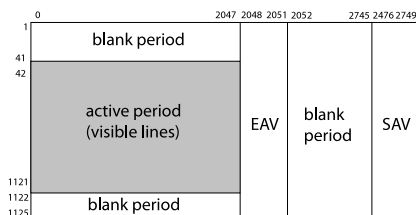


Fig. 3. Format of the 2K frame.

complete frame consists of 1125 lines (Fig. 3). The 1.485 Gb/s data stream is divided into 10-bit words. At 24 frames per second, there are two 10-bit words available for one sample. With 10-bit color depth and 4:2:2 subsampling, one word can carry the luma substream (Y') and the other word can carry the color substream ($C'_R C'_B$) [7]. Other color depth and subsampling options require the use of dual HD-SDI or 3G-SDI, both requiring twice the network bandwidth.

The advantage of this solution is that embedded audio and embedded data are included with no additional effort. The disadvantage is a high resulting bit rate. We can only transmit four HD-SDI channels over a 10 Gb/s network. Therefore, only 4:2:2 subsampling and 10-bit color depth is possible.

Just image & audio bits.

An alternative solution is to extract and transfer just the image and audio bits. One active line consists of 2048 image samples. One frame includes 1080 active lines. The number of bits per active sample depends on subsampling and color depth. When dual-link HD-SDI is used, one sample must be extracted from both channels, where it is distributed.

The advantage of this solution is a lower bit rate, which allows transmission of all subsampling and color depth options, including RGB at 12-bits per color. The disadvantage is more complex data transformation at both the sender and receiver. The embedded audio and embedded data also need to be extracted and transmitted by an additional mechanism.

Active area samples.

Another solution is to transfer HD-SDI data in its original format, but just for columns that include image samples, embedded audio or embedded data. Using this solution, we can transfer eight HD-SDI channels over a 10 Gb/s network with simpler data transformations on the sender and receiver. Still, all subsampling and color depth options are possible. This is the solution used in the current version of our platform.

Embedded audio can be located in up to 268 words of 10 bits per one line, including ancillary (blank period) lines [13], increasing the number of bytes transferable per line by 335. In the ancillary lines, just the embedded audio needs to be transferred.

The required number of bytes per line approximately corresponds with the maximum payload size of one Ethernet jumbo frame. Therefore, we chose to pack one line of one quadrant per one Ethernet frame.

Each frame adds an overhead of at least 46 bytes (8 bytes for UDP header, 20 bytes for IP header, 14 bytes for Ethernet header and 4 bytes for Ethernet CRC). More bytes may be required for Ethernet VLANs, MPLS, IP options or higher layer protocol headers.

In order to calculate bit rate at the physical layer, additional 20 byte intervals per packet need to be included for the Ethernet preamble and the inter frame gap (IFG).

Packet rate is approximately 100 000 packets per second and does not pose a problem in a 10 Gb/s network infrastructure. Distribution to multiple receivers is possible by IP or optical multicast. With IP multicast, we specify a multicast destination IP address and care should be taken to use an explicit join multicast protocol, such as PIM-SM [14]. An advantage of optical multicast [15] is independence of data rate and upper-layer technology.

Accidental flooding of a network by a high rate of data can be avoided by stopping data transmission when an ICMP destination unreachable message is received by the sender. Then, the data are only sent when a receiver is up and running at a specified IP address and expecting data at a specified UDP port.

3.3. Rendering rate adaptation

There are two sources of rate difference between the data arriving to the receiver from the network and the data to be sent to the rendering device. First, the internal clock of the sender can be different from the internal clock of the receiver, within a tolerance permitted by the transmission protocol. The HD-SDI clock rate is specified as 1.485 Gb/s, 1.485/1.001 Gb/s, 2.97 Gb/s or 2.97/1.001 Gb/s. Most devices indicate an acceptable tolerance of 100 ppm. Second, network delay variation [16] (jitter) can be introduced due to network traffic conditions. The jitter needs to be accommodated for by the receiver FIFO memory.

Several alternative techniques can be used to adjust the sending and rendering rate over an asynchronous network: receiver feedback, frame buffer, blank period adjustments, precise clock synchronization and rendering clock adjustments.

Receiver feedback.

The receiver can send feedback to the sender requesting sending rate adjustments (flow control). This technique is used in window-based transport layer protocols, such as TCP or in some link layer protocols, such as PAUSE frames in Ethernet.

The adjusted data rate appears at the receiver after round-trip time (RTT), which is typically hundreds of milliseconds on long-distance Internet connections. In high-definition video transfers, this technique would require a large receiver FIFO memory, which would introduce high latency. Therefore, we decided to use another technique.

Frame buffer.

This technique requires the receiver to have a large FIFO memory. Then, the rendering device can be driven by a fixed clock source in the receiver. The top of the FIFO memory holds the frame that is being sent to the rendering device.

After a long period when the skew between the sender and receiver clock rates causes the next frame to be rendered to not be completely available in time, the previous frame is rendered again. Similarly, if the FIFO cannot accommodate more incoming data, one frame is dropped. Of course, this solution causes a disruption to the audio signal transferred along with the video signal.

In the worst case when both the sender and receiver clocks are shifted by 100 ppm in opposite directions, the error can expand to the whole frame in $1/200 * 10^{-6} = 5000$ frames. At 24 frames

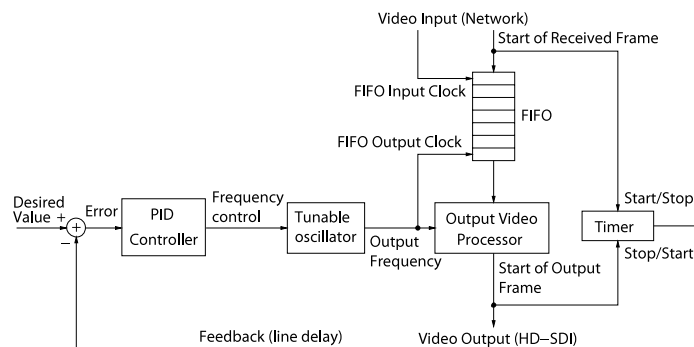


Fig. 4. Receiver control structure.

per second, it equals to approx. 3.5 min. A large FIFO memory can accommodate a shift by several frames, at the expense of latency.

We did not use this option because it would introduce high latency and problems with audio synchronization.

Blank period adjustments.

Most digital video transfer formats include a blank period in addition to an active period (visible lines) in each frame. The blank period is used for embedded audio, ancillary data and synchronization. The structure of the 2K frame [17] (or one quadrant of the 4K frame) as transmitted using an HD-SDI channel is shown in Fig. 3. Each frame consists of 1125 lines, which include 1080 active lines. Each line consists of 2750 samples, which include 2048 active samples. The active period is shown in gray. SAV (start of active video) and EAV (end of active video) are synchronizing sequences present in each line.

One simple technique to adjust the rendering speed to the rate of incoming data is to add or remove some samples or lines in the blank period. While this technique does not comply with SMPTE standards, we found that it works with some devices. Since we required compliance with standards, we also decided not to use this technique in our final solution.

Precise clock synchronization.

When both the sender and receiver are driven by a precise external frequency source, such as one locked on a PPS (pulse per second) signal from the GPS (global positioning system) receiver, the whole system can run stable for a long time. However, some senders cannot be synchronized to an external frequency source, such as medical equipment. Moreover, it is often difficult to obtain the GPS signal in locations where video signals need to be sent and received, for example lecture halls.

Rendering clock adjustments.

Adjusting the clock of HD-SDI channels between the receiver and the rendering device within the permitted tolerance gives the receiver some level of adaptation to the rate of incoming data. This solution requires a tunable oscillator and a closed-loop controller in the receiver. It is a solution that we implemented in our device.

3.4. Receiver controller

In order to adjust the rendering clock to the data source rate, we used a common PID controller [18]. The complete receiver control structure is shown in Fig. 4. The FIFO input clock is driven by data arriving from the network. The FIFO output clock is driven by a clock generator, which is tuned by the PID controller.

The bare adjusting of the rendering clock would stabilize the rendered picture, but it does not guarantee frame alignment. The top of the picture could appear anywhere on the screen.

Therefore, we used as the feedback variable to drive the controller the delay between the time when the first active line enters the receiver FIFO memory and the time when the receiver starts to send the first active line to the rendering device measured in clock cycles.

This delay is written to a register for each frame and sampled by the regulator periodically. The controller then uses a weighted moving average of the last eight samples. The purpose of this average is to smooth out fluctuations in FIFO occupancy due to network jitter.

One may argue why we do not use the time when the first active line leaves the FIFO instead. This would actually introduce instability, which we confirmed in our tests. If the FIFO occupancy increases due to the receiver clock being slower than the sender clock, packet loss at the FIFO input will increase as a result of network jitter. This will increase probability that the first active line (or any other line used by the regulator) is also lost. And this will cause the regulator to react more slowly, thereby allowing further increase in the FIFO occupancy and higher packet loss.

The feedback variable is compared with the desired value, whose value should be optimized empirically for maximum picture stability (see Section 4). The PID controller then produces adjustments to the clock generator frequency.

3.4.1. Out-of-range adjustments

The receiver FIFO can accommodate some fluctuations in data arrivals on its input end. But when a line is not available at the output end when it should be rendered, the receiver sends another line to the rendering device, because the signal on the synchronous HD-SDI channels cannot be stopped. This is either some line which happens to be at the FIFO output end or a copy of the previous line if FIFO is empty. When the newly arriving line does not fit into the receiver FIFO, it is dropped. These events cause the whole picture to roll up or down. The controller will see a large value of the error variable and try to compensate quickly.

4. Practical experience

We first tested our device in the laboratory by transferring patterns from the 2K generator over a short distance. The signal from the generator was distributed to all four quadrants. On the receiver side, an HD-SDI to HDMI converter and scaler with an HD monitor were used to check the signal.

890

J. Halák et al. / Future Generation Computer Systems 27 (2011) 886–892

```

raster= 779 avgf= 1034 speed_avgf= -45.20 freq=148496064 df= 21.26
raster= 1778 avgf= 1220 speed_avgf= 12.63 freq=148496048 df= -9.51
raster= 750 avgf= 1102 speed_avgf= -19.89 freq=148496048 df= 7.92

```

Fig. 5. Example of PID controller output.

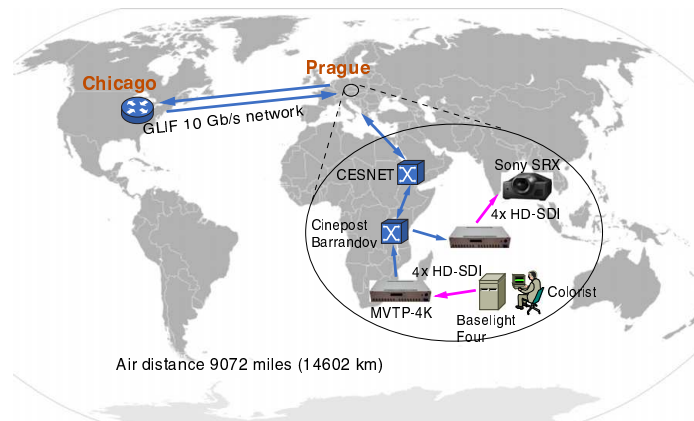


Fig. 6. Remote loopback test (Prague–Chicago–Prague loop, 14 602 km).

As the next step, we transferred the 4K signal over a long-distance loop from Prague to Chicago and back to Prague.

Finally, we demonstrated a real use of the technology for real-time remote color grading of uncompressed 4K video content between continents at the CineGrid 2009 workshop.

4.1. Laboratory tests

We implemented the proposed architecture on the Xilinx Virtex 5 FPGA device. The software part runs on embedded Linux, which runs inside the FPGA using the softcore Microblaze [19] processor. The Linux environment provides access to firmware variables and runs the PID controller process, which communicates with a tunable oscillator. The prototype version used an external laboratory clock generator with an RS-232 interface. The current version uses a built-in tunable oscillator. It is more convenient to generate commands for the tunable oscillator in software, rather than hardware.

Without a controller, the picture started to roll down or up after about 3–4 s even when communicating over a single 10-Gb Ethernet segment. This was due to the difference and wander of the sender and receiver clocks. When the controller was activated, the picture was stable and did not move by a single line after one day of continuous streaming. The controller reads the feedback variable, computes the error variable and generates commands for the clock generator periodically. In this way, it can stabilize the picture continuously. Most fluctuations in data arrivals were due to network jitter, which was handled by the FIFO memory.

A few lines of the PID controller output are shown in Fig. 5. Each line indicates one adjustment. The numbers shown from left to right: feedback variable in clocks, moving average of feedback variable in clocks, change of this average between two samples, resulting frequency to be set on the clock generator in Hz and change of this frequency in Hz.

4.2. GLIF loopback test

We tested our device over the GLIF (Global Lambda Interchange Facility) network in an L3 loop Prague–Chicago–Prague. The air

distance was approx. 9072 miles or 14602 km. The setup is illustrated in Fig. 6. In this configuration, network jitter was much higher and frequently exceeded the FIFO capacity. By duplicating or removing single rows complemented with rendering clock adjustments by the controller, the impairments were not subjectively observable.

4.3. CineGrid demonstration—remote color grading

At the CineGrid 2009 workshop, we demonstrated the use of the described technology for real-time remote color grading of uncompressed 4K video between continents over a distance of more than 6200 miles (10 000 km). CineGrid is a not-for-profit membership organization whose aim is to build a multidisciplinary community that promotes research, development and adoption of technologies for the exchange of high-quality digital media over high-speed networks.

The aim of the demonstration was the presentation of remote collaboration during the color grading process, where the grading system and its operator (the colorist) were in Prague, while the Director of Photography (DoP), who instructed the colorist what to do and checked the results, was in San Diego. The current state of the art in the movie industry is to have all persons in the same place. This leads to a non-effective allocation of resources during the post-production phase, where the key people are often highly distributed across continents and spend a lot of non-productive time while traveling.

The demonstration setup is illustrated in Fig. 7. The 4K content was streamed from the Baselight Four at the Cinepost corporation at Barrandov Studios in Prague. This content was transferred using two MVTP-4K devices over the GLIF network from Prague over Chicago to the University of California in San Diego (UCSD), where the CineGrid workshop took place.

The connection over the GLIF network consisted of a series of 10 Gb/s circuits inter-connected by an L3 router in Chicago and several L2 switches along the route. The used VLAN was not completely dedicated for the demonstration and there was a small volume of other background traffic.

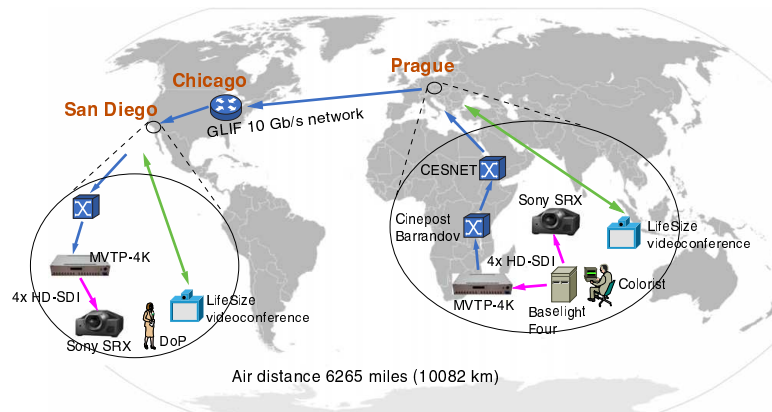


Fig. 7. Schematic diagram of the CineGrid 2009 demonstration.

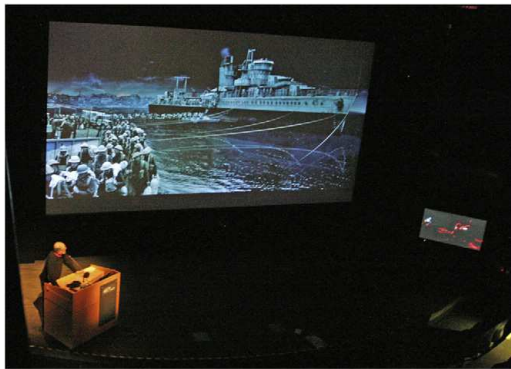


Fig. 8. CineGrid 2009 demonstration appearance.

Additionally, there was also a bidirectional LifeSize videoconference connection between Cinepost and the CineGrid venue. The director of photography (DoP) at the CineGrid workshop used this videoconference to discuss color grading of the 4K content with the colorist person at Cinepost, who performed these corrections in real-time on the 4K content streamed to the CineGrid venue. The overall appearance in the UCSD lecture hall is shown in Fig. 8.

5. Related work

Net Insight's [20] Nimbra 600 series switch can transport $8 \times$ HD-SDI or 3G-SDI channels over a synchronous SONET/SDH network. There are several commercially available solutions for transport of compressed 4K video over the Internet, for example NTT Electronics [21] ES8000/DS8000 4K MPEG-2 encoder/decoder complemented with NA5000 IP interface unit or intoPIX's [22] system of PRISTINE PCI-E FPGA boards and JPEG 2000 IP cores. The first demonstration of uncompressed 4K transmission over a long distance network was described in [23]. This system was based on PCs with HD-SDI cards with an additional PC on the receiver side for synchronization of the four channels.

The system described in this article can transport uncompressed 4K video streams over an asynchronous network with small latency (less than 1 ms added to the networking delay), it

is scalable to a higher number of transferred HD-SDI channels and operates bidirectionally.

6. Conclusion and future directions

We have demonstrated that it is possible to transfer uncompressed 4K video content over a long-distance network in real time without observable visual disturbances using FPGA technology for data framing, deframing and rendering speed control.

The technology enables distributed teams to significantly increase their productivity by sharing the uncompressed 4K video content in real time. The application areas include film post-production, scientific visualisation, art performances or high-quality videoconferences. The paradigm (division of participants) can also be used for other parts of the film (post-)production, including editing, special effects or dailies workflow.

In our further work we plan to support bidirectional transfers and explore what video processing functions would be useful to incorporate in the firmware, such as encryption or transcoding. An 8K uncompressed version over a 40 Gb/s network is technically possible if there is demand.

Acknowledgements

This work has been supported by the Ministry of Education, Youth and Sports of the Czech Republic under the research intent MSM6383917201 "Optical Network of National Research and its New Applications".

The tests and CineGrid demonstration using the Baselight Four and the Sony SRX 4K projector were possible thanks to the generosity of the Cinepost corporation.

References

- [1] L. Renambot, A. Rao, R. Singh, B. Jeong, N. Krishnaprasad, V. Vishwanath, V. Chandrasekhar, N. Schwarz, A. Spale, SAGE: the scalable adaptive graphics environment, in: Proceedings of WACE 2004.
- [2] Kevin Ponto, Kai Doerra, Falko Kuester, Giga-stack: a method for visualizing giga-pixel layered imagery on massively tiled displays, *Future Generation Computer Systems* 26 (5) (2010) 693–700.
- [3] Carolina Cruz-Neira, Daniel J. Sandin, Thomas A. DeFanti, Surround-screen projection-based virtual reality: the design and implementation of the CAVE, in: Proceedings of SIGGRAPH'93, pp. 135–142.
- [4] Thomas A. DeFanti, Gregory Dawe, Daniel J. Sandin, Jurgen P. Schulze, Peter Otto, Javier Girado, Falko Kuester, Larry Smarr, Ramesh Rao, The StarCAVE, a third-generation CAVE and virtual reality OptiPortal, *Future Generation Computer Systems* 25 (2) (2009) 169–178.

- [5] Charles Poynton, Chroma subsampling notation, in: *Digital Video and HDTV: Algorithms and Interfaces*, Morgan Kaufmann, 2002, (Chapter).
- [6] ITU-T recommendation G.114—one-way transmission time, ITU-T Study Group 12, May 2003.
- [7] 1.5 Gb/s signal/data serial interface, SMPTE 292-2008.
- [8] Dual link 1.5 Gb/s digital interface for 1920 × 1080 and 2048 × 1080 picture formats, SMPTE 372-2009.
- [9] Television-3 Gb/s signal/data serial interface, SMPTE 424M-2006.
- [10] Television-1920 × 1080 image sample structure, Digital Representation and Digital Timing Reference Sequences for Multiple Picture Rates, SMPTE 274M-2008.
- [11] Ultra high definition television-image parameter values for program production, SMPTE 2036-1-2009.
- [12] Interfaces for the optical transport network (OTN), ITU-T Recommendation G.709.
- [13] 24-bit digital audio format for SMPTE 292 bit-serial interface, SMPTE 299-2009.
- [14] B. Fenner, M. Handley, H. Holbrook, I. Kouvelas, Protocol independent multicast-sparse mode (PIM-SM): protocol specification, RFC 4601, IETF, 2006.
- [15] Yinzhu Zhou, Gee-Swee Poo, Optical multicast over wavelength-routed WDM networks: a survey, *Optical Switching and Networking 2 (3)* (2005) 176–197.
- [16] C. Demichelis, P. Chimento, IP packet delay variation metric for IP performance metrics (IPPM), RFC 3393, IETF, 2002.
- [17] D-cinema distribution master-image pixel structure level 3-serial digital interface signal formatting, SMPTE 428-9-2008.
- [18] Bela Liptak, *Instrument Engineers' Handbook*, third ed., Process Control, Butterworth-Heinemann, ISBN: 0801982421.
- [19] MicroBlaze soft processor core. <http://www.xilinx.com/tools/microblaze.htm>.
- [20] Net insight AB. <http://www.netinsight.se>.
- [21] NTT electronics. <http://www.ntt-electronics.com>.
- [22] IntoPIX. <http://www.intopix.com>.
- [23] Daisuke Shiraia, Tetsuo Kawano, Tatsuya Fujia, Kunitake Kanekob, Naohisa Ohtab, Sadayasu Onob, Sachine Araic, Terukazu Ogoshic, Real time switching and streaming transmission of uncompressed 4K motion pictures, *Future Generation Computer Systems 25 (2)* (2010) 192–197.



Jiří Halák received his M.Sc. at the Czech Technical University and is working towards his Ph.D. in Computer Science. He is also working as a researcher in CESNET, particularly in the field of programmable hardware.

Michal Krsek received his B.Sc. at the University of West Bohemia. He is currently with CESNET and his research interests include high-definition video and its applications.



Sven Ubik received his M.Sc. and Dr. in Computer Science from the Czech Technical University in 1990 and 1998, respectively. He is currently with the Research and development department of CESNET. His research interests include network monitoring, high-definition video, programmable hardware and optical networks. He is Cisco CCIE #14053.

Petr Žejdl received his M.Sc. at the Czech Technical University and is working towards his Ph.D. in Computer Science. He is also working as a researcher in CESNET, particularly in the field of programmable hardware.

Felix Nevřela is a Managing Director of the Cinepost corporation.

5.3 J. Halak, S. Ubik, and P. Zejdl. Scalable embedded architecture for high-speed video transmissions and processing

This paper was published in: *Proceedings of the Sixth International Conference on Systems and Networks Communications* in 2011 [A3]. The authors received The Best Paper Award for this paper.

This paper includes a grayed text that is not considered as a part of this doctoral thesis. The grayed text describes the research results proposed by paper co-author Jiří Halák and are considered as a part of his dissertation thesis. These results were only used and are not proposed by me.

ICSNC 2011 : The Sixth International Conference on Systems and Networks Communications

Scalable Embedded Architecture for High-speed Video Transmissions and Processing

Jiří Halák, Sven Ubik, Petr Žejdl
 CESNET / CTU Prague
 Žitkova 4, Prague 6 / Kolejní 550, Prague 6
 Czech Republic
 email: {halak,ubik,zejdl}@cesnet.cz

Abstract—In this paper, we present a scalable and extendable hardware architecture for processing and transfer of ultra-high-definition video over high-speed 10/40/100 Gbit networks with very low latency. We implemented this architecture in a single FPGA device. Data processing is divided between FPGA resources and an embedded operating system. The FPGA resources can be moved between various processing functions depending on the device mode. The resulting inexpensive and compact device is intended for high quality video transfers and processing with a low latency and to support deployment in education and remote venues.

Keywords—HD-SDI, video, FPGA, network communication, high-speed

I. INTRODUCTION

Video transfers are an expected driver application area of the future Internet. Picture resolution has been increasing over time. Better-than-high-definition-resolution video (such as 4K) is already used in some areas, such as scientific visualization, the film industry or even medical applications.

For the ultimate quality, required for instance in film post-production or live remote surgery transmissions, working with a signal that has not been compressed is preferable. The productivity of a distributed team can be significantly increased when the video signal can be transferred over the network in a real time to enable cooperation that is more effective. Two of the main technical issues are high-data volume and time synchronization when transferring over an asynchronous network such as the current Internet.

Currently available solutions mostly consist of multiple devices (computers, conversion boxes, sync boxes, audio boxes, etc.), which are expensive and harder to setup, increasing the logistics costs. We designed an embedded modular and scalable architecture which fits into a single mid-size FPGA device including all the required functionality and reducing the complexity and costs of this solution. We implemented this architecture and developed a device called MVTP-4K (Modular Video Transfer Platform). We have already used several prototypes in field tests to support applications in film post-production and live medical applications.

This paper is organized as follows: In Section II, we summarize the hardware requirements of our design. In Section III, we present our architecture for video transfers and processing. In Section IV, we present our prototype. In Section V, we summarize our experience with device field tests, In Section VI, we compare our solution with other available devices.

II. REQUIREMENTS

We have set the following set of requirements for our architecture:

- Video inputs and outputs SDI, HD-SDI or 3G channels
- 10/40/100 Gbit network interface or multiple interfaces
- Very small added latency
- Extendable design for additional processing such as compression or encryption
- Fit into available FPGA devices and fully implementable in one mid-size FPGA device with additional interfaces

The use of a single- and dual-link HD-SDI channel for the transmission of high definition video streams is now a common industry practice and it is specified in SMPTE 274 [1] and SMPTE 372 [2]. This includes HD (1920x1080) and 2K (2048x1080) formats. The 4K (4096x2160) signals are typically transferred in four quadrants, each in 2K format carried over a separate dual-link HD-SDI channel. 3D transmissions are typically transferred as two independent 2K or 4K channels, some require additional synchronization.

The FPGA circuit was chosen as the processing device due to its versatility that allows us to build a complete embedded solution and to host all required functionality to combine video transmissions with other functions, such as compression, encryption or transcoding.

The architecture must be scalable to allow multiple configurations based on currently available FPGA devices and interfaces, assuming the speed of communication interfaces will increase, and eventually be usable with future 100 Gigabit Ethernet networks and similar high-speed media.

We require an unnoticeable latency added to the network propagation delay for real-time applications. Unnoticeable

latency for audio/video applications is below 60 ms for un-trained audience and below 30 ms for professional audience.

III. THE ARCHITECTURE

This section describes the proposed architecture.

A. Background

In our previous work we designed and implemented a scalable hardware architecture for network packet processing [3], [4]. This architecture consists of a set of reconfigurable modules for packet processing and a communication interface. The architecture was designed for maximum flexibility and multi-gigabit speeds starting at 10Gb/s. The main processing core was designed to be fully scalable for 10/40/100Gb speeds.

We have designed an interface and developed a prototype for 40Gb/s SONET/SDH networks [5] for basic data processing and testing of 40Gb/s networks and currently we are experimenting with 100Gb Ethernet interface in FPGA devices.

B. Design Overview

Real-time processing of multi-gigabit data rates is difficult on PC-based platforms with standard operating systems not designed for real-time operation. We were looking for a real-time design that is scalable to higher data rates (such as for 8K or UHD TV2 format), higher network speeds (such as 40 and 100 Gb/s) and can be integrated with commonly requested video processing functions, such as encryption, transcoding or compression. Real-time operation means to add a very low latency to a network delay and enable true live experience. This design is fully automated and all embedded in a single FPGA device.

The embedded architecture for real-time video transport and processing is based on our previous work. The core is the scalable architecture for network packet processing [3], [4] designed especially for Ethernet networks. This whole architecture operates at network clock domain of attached network interface and can be used for various modular data packet processing. Since video signal consists of special packets, we can make a simple conversion to transport the video packets to a network clock domain and back. This way we can use a network packet processing architecture for video packet processing.

When we convert data packets from network clock domain to video clock domain certain mechanisms need to be used. Ethernet Network is an asynchronous network, on the other hand, HD-SDI is a synchronous channel thus an advanced techniques for synchronization of data packets crossing from network domain to video domain are required [6].

Address range of all processing modules, routes and hardware configuration registers is mapped to an embedded processor logic bus (PLB) address range using a simple bus bridge of our own design. An embedded processor can be a

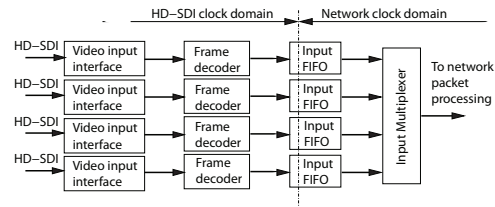


Figure 1. Input video processing and connection to packet processing for 4 channels.

dedicated Power PC processor or soft-core Microblaze [7] processor. An embedded processor is running a customized Linux distribution and all peripherals are managed by Linux drivers or dedicated software tools. The embedded Linux distribution also provides all means of communication with a device, such as ssh server, web server, display and keyboard controllers and eventually can also handle 10/100/1000 Mbit and multi-gigabit interfaces. The Multi-gigabit Ethernet Network Interface for an embedded processor is described in a subsection III-F.

C. Video Processing Modules

Video processing modules do a conversion between video and network packets, allowing video data to be processed in the network packet processing core (section III-D). There are two video processing modules, the input and output module, shown in Figures 1 and 2.

The input module consists of the video input interface and the frame decoder. The video input interface implements low-layer communication with the HD-SDI equalizer chips through Rocket IO channels and the frame decoder extracts video packets, converts them to network packets and attaches headers with video format parameters.

The output module consists of the video frame generator and the video output interface. The frame generator receives network packets and generates valid image to the video output interface based on information contained in network packet headers.

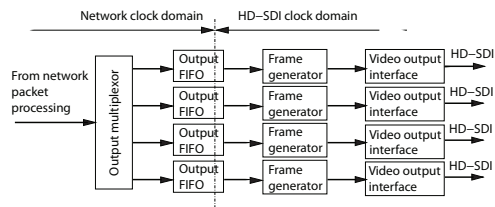


Figure 2. Output video processing and connection to packet processing for 4 channels.

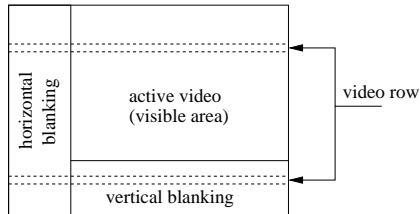


Figure 3. Video frame structure transported through HD-SDI channel

A video packet includes a video pixel row with specified headers and control characters. We use a dual-port memory as a packet FIFO to cross clock domain boundaries. The video processing modules are located in the HD-SDI clock domain and the network packet processing modules are located in the network clock domain. The example configuration in Figures 1 and 2 includes four HD-SDI channels. The channels are independent and can be added freely just with a simple modification of the channel multiplexer. This operation can be completely parameterized.

The HD-SDI interface has a bit rate of 1.485 Gbit/s [8] but not all data needs to be transferred. Video rows include blanking areas (horizontal blanking interval) and a video frame includes blanking video rows (vertical blanking interval). The whole situation is illustrated in Figure 3. Blanking areas can contain some secondary information such as audio, encryption or video format specification, which we can choose to transport or not. When we strip video packets of those blanking intervals, we get a bit rate between 1 Gb/s and 1.3 Gb/s depending on a picture resolution and frame rate. This means that the 10 Gbit Ethernet network can transfer up to eight HD-SDI video channels and with some video formats even including additional data, such as audio or encryption information.

The example bitrates of eight channels of selected video formats stripped of blanking intervals are summarized in Table I. The 30fps HD formats can be still transferred, but image crop must be applied.

TABLE I
VIDEO FORMATS BITRATES

Format	Bitrate eight channels (Gb/s)	Bitrate one channel (Gb/s)
2K/24	8,7	1,08
1080/24	8,2	1,025
1080/25	8,5	1,06
1080/30	10,4!	1,3
720/50	7,6	0,95
720/60	9,1	1,14

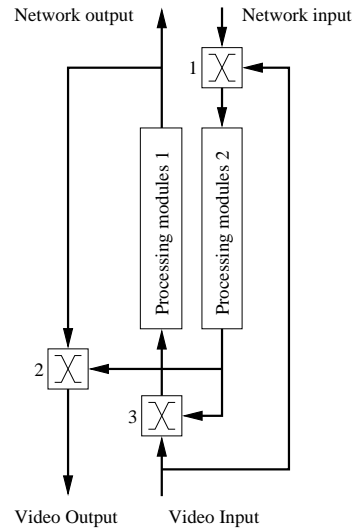


Figure 4. Schematic of interconnections in the processing core.

D. Network Packet Processing Core

The main processing core consists of two sets of processing modules. We have extended our original architecture [3], [4] with the video interface and video processing modules described in section III-C. The network packet processing core is divided into two parts. A set of switches can be arranged to allow a packet flow between the network and video domains in several ways. A schematic of this interconnection is shown in Figure 4. The following configurations are possible:

- From network input to network output through switch 1, processing modules 2, switch 3 and processing modules 1. All processing modules are dedicated for network to network packet processing.
- From network to video, full-duplex, one set of processing modules for each direction. From network input through switch 1 and processing modules 2 to video output. From video input through switch 3 and processing modules 1 to network output.
- From video input to video output through switch 3, processing modules 2, switch 3, processing modules 1 and switch 2. All processing modules are dedicated for video packet processing.

Data stream processing modules are inserted directly to the packet stream. Every processing module works as an individual processing unit. The advantage is that modules can occupy different FPGA devices. When we need to implement a complex module such as video encoder/decoder,

we may find more suitable to use more FPGA devices. For this purpose, the architecture is designed to relocate a packet stream through a high-speed FPGA ports to another device and make a cross-device interconnection of processing modules. Both options are shown in Figure 5. Option A: Modules are connected in a single device. Option B: Module interconnection crosses multiple devices over a high-speed interface.

E. Processing latency

The concept of intra-frame processing of video packets as network packets enables extra low latency of video processing and transmission. This opens a way to truly real-time collaboration support. The processing design itself has a low latency under 1 ms. Video packets are buffered only when synchronizing from the network asynchronous domain to the video synchronous domain. However, low delay variation in the network is required to allow design latency under 1 ms. In lower quality networks the buffering level needs to be obviously increased. The extreme cause is a single frame buffer adding a maximum latency of about 30 ms.

F. Network Interface

High-speed network interface consists of hardware and software parts, which are controlled by an embedded operating system. Incoming packets containing video data are sent to output video processing module, on the other hand network management packets such as ARP or ping are sent to software network driver. Outgoing packets have two different sources, packets containing video data are sent from input video processing module and network management packets are sent from software network driver.

The block diagram of the network interface is shown in Figure 6. Incoming packets are classified in the packet classifier and distributed between video processing modules (VPM) and RX FIFO. Outgoing packets are sent from VPM or from TX FIFO. Because there are two paths producing packets, packet multiplexer is included in the design. It is

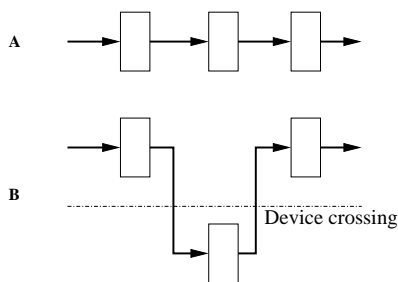


Figure 5. Processing modules

multiplexing packets in a round-robin fashion. RX and TX FIFO are connected to the CPU through the processor local bus. Therefore, both memories are accessible from software. The packet classifier is also connected to the CPU, but the connection is not shown. The CPU is embedded inside FPGA either.

The packet classifier contains memory for four classification rules. Each rule can be marked as going to the VPM and/or going to RX FIFO. The memory is configured from software. Currently we use three rules. The first rule is marked as going to the VPM and the other two rules are marked as going to RX FIFO. The fourth rule is not used and is reserved for future use.

The rules are as follows:

- Rule for UDP packets containing video data.
- Rule for ARP packets for address resolution.
- Rule for ICMP packets (ping command).

The software part is based on embedded Linux. Network interface is accessible through the Linux TUN/TAP driver [9], which provides packet reception and transmission for user space programs. The program controlling network hardware is running as a daemon in the user space, and through TUN/TAP driver provides a new network interface. This new interface behaves like an ordinary network interface such as eth. Therefore, all networking services are available through this interface.

IV. MVTP-4K PROTOTYPE

We have designed and build a MVTP-4K (Modular Video Transfer Platform) device which implements proposed architecture and validates it in field tests. The MVTP-4K is a

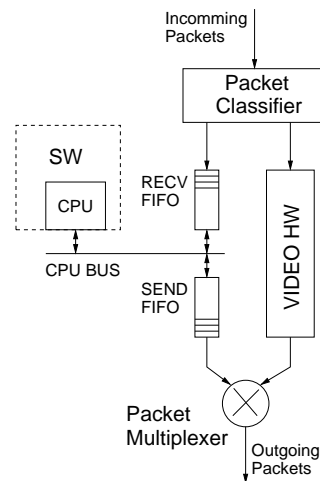


Figure 6. Ethernet Interface Block Diagram

ICSNC 2011 : The Sixth International Conference on Systems and Networks Communications



Figure 7. System architecture overview.

portable device of our own construction for transmission of multiple high-definition video streams including 4k, 2k and HD over a 10 Gigabit Ethernet Network. The device consists of a main FPGA board with 8 HD-SDI video interfaces and one 10Gbit Ethernet interface. Brief structure is shown in Figure 7. The device supports all 4K, 2K and HD resolutions and all corresponding frame rates. 3D transmissions are also supported. Because the data processing is based on data packet processing we can even transport data not fully supported without the need of unpacking them from video signal. This allows us to transport audio data or encryption data embedded in the video stream.

We have chosen a Xilinx Virtex FPGA series because it provides all building blocks and tools required to implement our architecture. The prototype is based on an extended platform for network packet processing called MTPP [3]. Whole architecture fits to a mid-size FPGA device Virtex 5 series XCV5LX110T. We have experimentally confirmed that the device adds a low latency of less than 1 ms.

Mid-size Xilinx FPGAs can be obtained under 3000\$ a piece in a small quantities. For advanced hardware functions such as encryption or encoding, a larger FPGA or a second mid-size FPGA is required.

V. PRACTICAL EXPERIENCE

We have demonstrated our system at the Cinegrid 2009 and Cinegrid 2010 workshops. The aim was to demonstrate that such technology can enable real-time remote cooperation of a distributed team and thus increase productivity. In the first event, a stream of uncompressed 4K video was



Figure 8. Practical use of the technology at Cinegrid 2010 event

transferred from the Barrandov studios in Prague to the venue in San Diego over a distance of more than 10000 km to perform remote color grading in a real-time. In the second event, a stream of 3D 2K video was transferred from the UPP Company in Prague to the venue in San Diego to perform remote real-time postproduction processing of 3D images. The 3D grading performed at the venue with the signal transferred by our device is illustrated in Fig.8.

In order to evaluate the system suitability for e-Health applications, we transferred several surgical operations from the daVinci Surgical System [10] which produces HD stereoscopic signal in 1080i format. The picture quality was subjectively approved by invited medical experts as suitable for highly illustrative student training or presentations of surgical procedures on symposia.

VI. RELATED WORK

There are several commercial products, which allow transport of SDI, HD-SDI or 3G channels over network.

Net Insight's [11] Nimbra 600 series switch can transport 8x HD-SDI or 3G SDI channels over an SONET/SDH network. There are several commercially available solutions for transport of compressed 4K video over the Internet, for example NTT Electronics [12] ES8000/DS8000 4K MPEG-2 encoder/decoder complemented with NA5000 IP interface unit and intoPIX's [13] system of PRISTINE PCI-E FPGA boards and JPEG 2000 IP cores.

UltraGrid from Laboratory of Advanced Networking Technologies is a software for real-time transmissions of high-definition video [14]. This solution is a fully software based and requires dedicated PC with specialized hardware.

The architecture and design described in this article differs in that it is a hardware solution fully scalable to higher speeds. The number of video and network interfaces is parameterized and can be easily extended. The FPGA enabled parallelism allows our architecture to process several video channels at once and to transfer every video format contained in SDI, HD-SDI or 3G interface. The architecture is designed to be embedded to a single FPGA device but some larger processing modules can be relocated to other FPGA devices. Our design has a very small added latency around 1 ms that enables a true real-time distributed team cooperation.

VII. CONCLUSION

We have extended a scalable architecture for network packet processing [3], [4] by video interfaces options. The resulting architecture is designed to process or transport video data over an asynchronous network with very low added latency. The design enables true real-time distributed team cooperation. The real-time team cooperation was demonstrated in several applications in the cinema industry and e-Learning in medicine. The architecture also fulfills the hardware requirements that we set and we successfully

implemented this architecture in a single FPGA device and presented its capabilities.

ACKNOWLEDGMENT

This work has been supported by the Ministry of Education, Youth and Sports of the Czech Republic under the research intent MSM6383917201 Optical Network of National Research and Its New Applications.

REFERENCES

- [1] "1920 x 1080 Image Sample Structure, Digital Representation and Digital Timing Reference Sequences for Multiple Picture Rates." Society of Motion Picture and Television Engineers., 2005.
- [2] "Dual Link 1.5 Gb/s Digital Interface for 1920 x 1080 and 2048 x 1080 Picture Formats." Society of Motion Picture and Television Engineers., 2009.
- [3] J. Halak and S. Ubik, "MTPP - Modular Traffic Processing Platform," in *12th IEEE Symposium on Design and Diagnostics of Electronic Systems, DDECS 2009*, Liberec, Czech Republic, April 2009, pp. 170–173.
- [4] J. Halak, "Multigigabit network traffic processing," in *Proc. The International Conference on Field Programmable Logic and Applications, FPL 2009*. Prague, Czech Republic: IEEE Computer Society, August 2010, pp. 521–524.
- [5] J. Halak, S. Ubik, and P. Zejdl, "Data stream processing for 40 Gb/s networks," in *Proc. Fifth International Conference on Digital Telecommunications, ICDT 2010*. Athens/Glyfada, Greece: IEEE Computer Society, June 2010, pp. 149–152.
- [6] —, "Receiver synchronization in video streaming with short latency over asynchronous networks." Vienna, Austria: IEEE Computer Society, April 2010, pp. 403–405.
- [7] MicroBlaze Soft Processor Core. (Last accessed: July, 2011). [Online]. Available: <http://www.xilinx.com/tools/microblaze.htm>
- [8] "1.5 Gb/s Signal/Data Serial Interface." Society of Motion Picture and Television Engineers., 2008.
- [9] Universal TUN/TAP device driver, Linux Kernel Documentation. (Last accessed: July, 2011). [Online]. Available: <http://kernel.org/doc/Documentation/networking/tuntap.txt>
- [10] The da Vinci Surgical System, Intuitive Surgical. (Last accessed: July, 2011). [Online]. Available: <http://www.intuitivesurgical.com/products/faq/index.aspx>
- [11] Net Insight AB. (Last accessed: July, 2011). [Online]. Available: <http://www.netinsight.se>
- [12] NTT Electronics. (Last accessed: July, 2011). [Online]. Available: <http://www.ntt-electronics.com>
- [13] intoPIX. (Last accessed: July, 2011). [Online]. Available: <http://www.intopix.com>
- [14] P. Holub, L. Matyska, M. Liska, L. Hejtmánek, J. Denemark, T. Rebock, A. Hutánu, R. Paruchuri, J. Radil, and E. Hladk, "High-definition multimedia for multiparty low-latency interactive communication," *Future Generation Computer Systems*, vol. "22", no. "8", pp. "856 – 861", October "2006". [Online]. Available: "<http://www.sciencedirect.com/science/article/pii/S0167739X06000380>"

5.4 S. Ubik, J. Navratil, P. Zejdl and J. Halak. Real-Time Stereoscopic Streaming of Medical Surgeries for Collaborative eLearning

This paper was published in: *Proceedings of the 9th International Conference on Cooperative Design, Visualization, and Engineering* in 2012 [A4].

Real-Time Stereoscopic Streaming of Medical Surgeries for collaborative eLearning

Sven Ubik, Jiří Navrátil, Petr Žejdl, and Jiří Halák

CESNET, Zikova 4, Prague 6, Czech Republic

Abstract. Medical surgeries in various specialities have been recently enhanced by modern devices with 3D vision for the surgeon. By transferring this 3D vision in high quality and with low latency to distant locations, we can enable novel collaborative teaching programs for medical students and doctors, also allowing remote interaction with the surgeon. We describe our experience with real-time long-distance stereoscopic transmissions of medical surgeries using a system for low latency streaming over packet networks. We discuss options for 3D transmission, 3D projection and experience of users that took part in multiple demonstrational transmissions.

Keywords: robotic surgery, eLearning in medicine, collaborative teaching, 3D transmissions

1 Introduction

Robotic surgery [1], such as using the da Vinci Surgical System ¹, brings several advantages to modern surgery techniques - precision, smaller incisions, decreased blood loss and consequently quicker healing time.

A stereoscopic camera is used to provide the surgeon with a view of the surgical elements. The signal from this camera can also be used for eHealth applications, such as remote medical students training or presentations of surgical procedures on symposia.

We need to transfer the stereoscopic high-resolution signal from the surgery device to the audience over potentially large distances in high quality and with short latency. The latter is important to provide interactive experience, where people in auditorium can ask questions and learn from the surgeon responses.

We describe our experience with using long-distance stereoscopic transmissions of high-definition vision of surgical procedures for eLearning in medicine.

2 Architecture

We have developed a device called MVTP (Modular Video Transmission Platform) [2]. It can transport bidirectionally up to 8 high-definition video channels

¹ <http://www.intuitivesurgical.com>

2 Sven Ubik, Jiří Navrátil, Petr Žejdl, and Jiří Halák

with very low latency. The processing delay of the sender and receiver together is less than 1 ms. If high network jitter is present, buffering must be configured to compensate, adding delay of tens of milliseconds. Network propagation delay ranges from about 20 ms across Europe to 150 ms from Europe to Japan.

Two channels can be used for the stereoscopic transmission of the operation. The third channel can be used in place of the video-conferencing system between the surgeon and people in the meeting room, including embedded audio. Other channels can be used to connect multiple operation devices, if more surgeries are to be presented during one event. A typical setup is shown in Fig. 1.

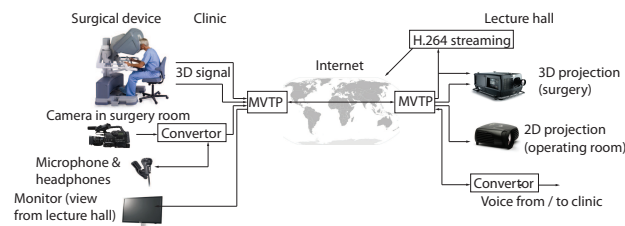


Fig. 1. Typical setup for remote collaborative eLearning using surgery images

3 Practical experience

One option to present an operation is to make a local projection near the operating room, see Fig. 2. Obviously the number of people that can be present there is very limited. Even if the projection is extended to a hospital meeting room, the medical experts or students would need to travel to the hospital.



Fig. 2. Local projection in a hospital

We arranged several long-distance transmissions for doctors, medical personnel and students. Some transmissions were in real time, during the operation. Other transmissions were pre-recorded. Real-time transmissions give a complete picture including the pre-operation phase. Pre-recorded transmissions allow to select particular operation phases and present them in shorter time.

We transmitted several urological operations done using the da Vinci Robotic System in cooperation with the Masaryk Hospital in Usti nad Labem in Czech Republic. For example, to the 5th Congress of Miniinvasive and Robotic Surgery in Brno, Czech Republic in 2010 or to the medical section of the APAN meeting in Hong Kong in 2011. The distances ranged from 300 km to 15000 km. The schematic diagram of the transmission to the APAN meeting is shown in Fig. 3.



Fig. 3. Transmission from Czech Republic to the APAN meeting in Hong Kong

The feedback from participants in the long-distance events was very positive. Medical experts and students found it very useful and educative to see the operations without the need to travel long distances. They particularly appreciated the collaborative nature of long-distance discussions with the surgeon. People asked impromptu questions, which were immediately answered by the surgeon. Low latency of the transmission system allowed for collaborative feeling. The stereoscopic projection also provided immersive feeling and increased the educative value the seen content.

Although bidirectional sound would be sufficient for communication, the surgeons appreciated the backward video stream from the venue as keeping better contact with participants. We found that the third video channel of the surgeon or the operating room is better to be shown on a second screen, see Fig. 4, rather than being switched with the stereoscopic operation on the main screen, which was confusing for some participants.

Technologies that have been used by other teams for medical transmissions include Skype and DVTS+ [3] for projection in a room and the *Connect for the da Vinci Si System* for transmission to a remote laptop. Our system adds 3D visualization and always used HD (1080 lines) resolution.

4 Sven Ubik, Jiří Navrátil, Petr Žejdl, and Jiří Halák



Fig. 4. Transmission to the medical seminar in Banská Bystrica in Slovakia

4 3D projection

There are several known options for 3D visualization. It can be shown with one frame-interlaced projector, two projectors for the left and right eye or a 3D LCD panel. Active or passive polarizing or anaglyph glasses can be used. We have tried various configurations and our recommendations are summarized in Table 1.

Number of participants	Method	Notes
up to 10	3D TV for passive glasses	least expensive, passive glasses are cheaper and do not need battery
10 to 20	projector for active glasses	bigger screen than TV, least expensive projector, normal screen can be used
20 to 50	projector for passive glasses	projector needs active filter, which increases cost, but it is offset by cheaper passive glasses and no hassle with batteries, for polarising glasses silver screen is needed
more than 50	cinema grade projector for passive glasses	expensive high-brightness projector (>10000 lumens) is needed for a large auditorium

Table 1. 3D visualization options

Two projectors have higher brightness, but are difficult to set up. If polarization filter is at projector (for passive glasses), polarization-preserving (silver) screen is needed. More expensive projectors have two video inputs for the left and right channel and multiplex these channels inside the projector. Cheaper projectors have one input and require an external multiplexor.

5 Conclusions and future work

We have shown that real-time long-distance transmissions of medical images can be used for interactive eLearning. Although the positive feedback is based on subjective experience of participants, we believe that it brings benefits to professional education.

We are installing a permanent 3D transmission and projection infrastructure between a local hospital and a medical faculty to support courses.

We plan a more rigorous evaluation of effect on learning quality as well as comparison of 3D and 2D transmissions. This will be done through questionnaires and capturing collaborative communication during courses.

We also plan experiments with a 3D projection of images obtained from a computer tomography for collaborative learning. Images from all cameras in a tomograph will be recalculated for a single wall or multiple wall 3D projection in the i CAVE (CAVE Automatic Visualization Environment) [4].

Acknowledgments This work was supported by the CESNET Large Infrastructure project (LM201005) funded by the Ministry of Education, Youth and Sports of the Czech Republic and by the POVROS project (TA01010324) funded by

References

1. Gharagozloo, F., Najam, F.: *Robotic Surgery*, McGraw-Hill, ISBN 007145912X.
2. Halak, J., Krsek, M., Ubik, S., Zejdl, P., Nevrela, F., Real-time long-distance transfer of uncompressed 4K video for remote collaboration, in *Future Generation Computer Systems*, vol. 27(7), pp. 886-892, doi:10.1016/j.future.2010.11.014.
3. A. Ogawa, K. Kobayashi, K. Sugiura, O. Nakamura, J. Murai, *Design and Implementation of DV based video over RTP*, IEEE Packet Video Workshop, Cagliari, Italy, 2000.
4. Cruz-Neira, C., Sandin, D.J., DeFanti, T., Kenyon, R.V. and Hart, J.C., The CAVE: Audio Visual Experience Automatic Virtual Environment, in *Communications of the ACM*, vol. 35(6), 1992, pp. 64-72, doi:10.1145/129888.129892.

5.5 P. Zejdl, S. Ubik, V. Macek, and A. Oslebo. Traffic classification for portable applications with hardware support

This paper was published in: *Proceedings of the International Workshop on Intelligent Solutions in Embedded Systems* in 2008 [A5].

Traffic Classification for Portable Applications with Hardware Support

Petr Žejdl, Sven Ubik, Vladimír Macek
CESNET

Arne Øslebø
UNINETT

Abstract

Traffic filtering and classification is needed in many monitoring applications. To process large volumes of data, we need hardware support embedded in monitoring cards. The problem is that different cards have different resources for filtering and classification.

We propose an architecture that enables utilization of available hardware resources in different monitoring cards and which can be easily extended to support future types of monitoring cards.

Consequently, monitoring applications can run transparently and efficiently in different hardware and software environments of current and future monitoring devices.

1: The Problem

Traffic filtering and classification is needed in many monitoring applications. For example, to reduce or distribute the volume of traffic to multiple processes or to compute statistics about specific traffic sources and destinations.

In *filtering*, we reduce the volume of traffic, while in *classification*, we mark packets as belonging to classes, which are then treated separately, for statistics or performance reasons.

Monitoring applications often need to run in varied hardware and software environments, such as different network link speeds, link types, network cards and operating systems. Filtering and classification of large volumes of data require support of hardware devices, such as FPGA or network processor-based monitoring cards. Examples of such devices are DAG [1] cards or COMBO [2] cards.

Therefore, there are two primary issues that need to be resolved:

1. Traffic filtering and classification should be portable to various hardware and software environments with minimal effort.
2. Hardware support should be utilized to the extend possible with the used monitoring cards and application requirements. Software replacement should be used otherwise.

2: Contributions

In this paper we propose a generalized architecture of traffic filtering and classification that provides the following benefits:

- Applications can run completely transparently in different hardware and software environments.

- Dynamic libraries provide hardware acceleration with various monitoring cards.
- Filtering and classification specifications use abstract data structures applicable to current and future hardware.
- The architecture is easily extendable with replaceable backends to support future hardware.

The main added value of the work described in this paper is that monitoring applications using MAPI (described in the next section) can now have their filtering and classification requirements accelerated using various monitoring cards transparently to the application.

For example, we may have application to measure distribution of network traffic into protocols. For a single wireless network, inexpensive hardware with a regular Ethernet card and libpcap library to capture and classify packets can be used. On the other hand, to monitor a high-speed trunk link, we need support of a specialized monitoring card.

3: Architecture

We describe the ideas of the proposed architecture in the following sections and we illustrate it in examples.

3.1: Filtering and classification transparency

An application programmer needs to specify filtering and classification using some notation and semantics. We decided to use the commonly known BPF (Berkeley Packet Filters) used by the libpcap library [3]. This allows easy porting of third-party libpcap-based application. We comment more on this decision in section 5.

For distribution of packets into classes and for transparency of monitoring cards to applications, we use the concept of replaceable dynamic libraries in MAPI [4] middleware, which is a de facto standard for development of passive monitoring applications. We added to MAPI support for transparent hardware-accelerated filtering and classification.

MAPI allows multiple applications to run concurrently over the same set of network cards. When a new application is started, its filtering and classification requirements are passed to MAPI middleware. These new requirements are added to requirements of already running applications. For each type of a monitoring card used by the new application, the dynamic library responsible for communication with that type of monitoring cards evaluates what filtering and classification could be hardware-accelerated and what need to be implemented in software. This is done based on the number and order of filtering and classification requirements and on the hardware resources in the given type of a monitoring card.

Those filtering and classification requirements that fit into hardware resources are passed to the translation system. Some structures however cannot be hardware-accelerated. For example, when packet payload bytes should be compared to specified values or when a dynamically determined offset in a packet header is referred to. In that case the translation system returns an error code and the MAPI middleware uses software implementation instead.

The architecture is illustrated in Fig. 1. The initialization and packet processing works as follows:

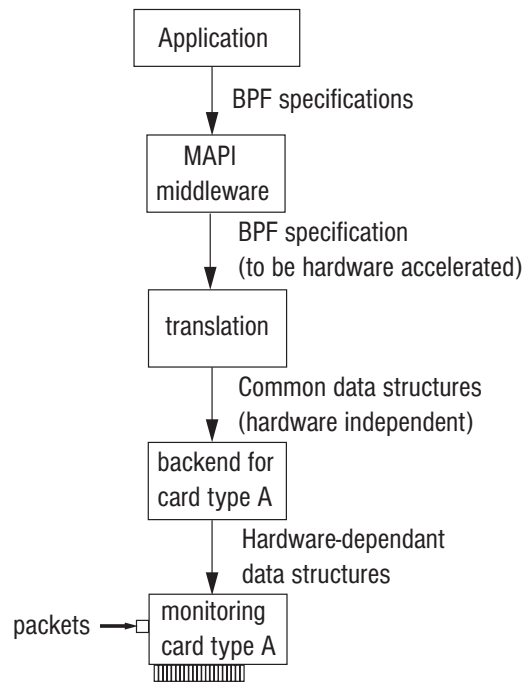


Figure 1. Architecture of transparent and extensible packet classification

1. Application uses `mapi_create_flow()` function to create one or more *flows*. Each flow is initially all packets arriving to one or more specified network interfaces.
2. Application then applies BPF strings to flows using `mapi_apply_function()`, thus implementing filtering and classification (depending on further packet processing).
3. Each type of supported network cards has a dynamic library implementing filtering and classification for that card (all standard Ethernet NICs are implemented as one type, specialized monitoring cards are other types).
4. An application uses `mapi_connect()` function to connect to MAPI middleware. At this point, MAPI makes the above described decision what BPF filtering and classification will be offloaded to hardware and what will be done in software.
5. If hardware support is used, monitoring cards are configured to filter and classify packets.
6. As packets arrive from network cards to the MAPI middleware, they are either filtered or classified in software or no action is performed when filtering or classification was already done in hardware.
7. When the application needs results of classification, they are taken from software or hardware counters.

3.2: Common data structures and translation process

Libpcap library compiles BPF strings into a register-based instructions and uses interpreter to execute them for each incoming packet. This allows early decisions about packet

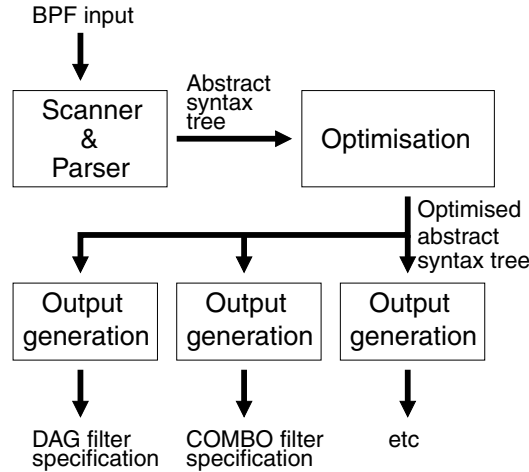


Figure 2. Translation of classification conditions

```

expr: term
    | expr and term
    { $$ .b = new_and_node($1.b, $3.b);
      $$ .q = $3.q; }
...
other: VLAN pnum
    { $$ = gen_vlan($2); }
    | VLAN
    { $$ = gen_vlan(-1); }
  
```

Figure 3. Example grammer rule

rejection when a part of a complex condition is false without evaluating the rest of the condition.

However, the interpreter approach is slow for large volumes of packets. Moreover, the libpcap library can only filter packets. In order to classify them, individual classification conditions need to be interpreted for each packet sequentially until the packet is classified.

Monitoring cards use hardware resources such as look-up tables and CAMs (Content Addressable Memory) that need a different type of filtering and classification implementation.

Therefore, we designed an abstract representation of classification conditions and the corresponding translation process keeping in mind hardware resources of monitoring cards, application transparency and extensibility for future hardware monitoring cards.

Structure of the translation process is depicted in Fig. 2. Two example translation rules are shown in Fig. 3. The input language is described by a context-free grammar using BNF (Backus-Naur-Form) notation annotated with semantic actions in C language in curly braces. Identifier `$$` is a return value of a rule. It may represents a single variable, structure or union. Identifiers `$$.b` refer to a binary tree, `$$.q` is the direction qualifier. Identifiers `$1`, `$2` and `$3` refer to the semantic values of the particular rule components.

- Syntax parser is implemented using *bison* and *flex* [5] tools.
- The abstract syntax tree uses several node types, see Table 1 chosen to represent

Node type	Information
binary-node	AND or OR operator
link-node	link type (IP, ARP, etc.)
host-node	IP address and netmask
protocol-node	network protocol (TCP, UDP, SCTP, etc.)
port-node	port number
port-range-node	range of port numbers

Table 1. Node types of internal tree representation

entities that can be usually evaluated by embedded logic in monitoring cards.

- Optimisation removes sub-expressions that can never match and converts the tree representation into DNF (Disjunctive Normal Form) [10].

The syntax parser accepts the following keywords:

- type qualifiers – `host net port portrange`
- direction qualifiers – `src dst`
- protocol qualifiers – `proto arp rarp vlan mpls ip tcp sctp udp`
- operators – `and && or || ()`

These qualifiers allow to create most expressions with common network protocols, which can be evaluated statically, that is using comparisons only. Such expressions can be successfully translated into configuration of look-up tables and CAMs in monitoring cards. For example, the following are statically evaluated expressions:

```
"tcp port 80"
"ip host www.google.com and \
  tcp dst port 80"
```

However, the following expressions includes arithmetics that requires runtime evaluation (it filters all IPv4 HTTP packets to and from port 80 that contain data, not SYN, FIN and ACK-only packets):

```
"tcp port 80 and (
  (ip[2:2] - ((ip[0]&0xf)<<2)) - \
  ((tcp[12]&0xf0)>>2) \
) != 0 \
)"
```

Possible evaluation in hardware would require some sort of arithmetic coprocessor. This is not the case with the supported monitoring cards, therefore this expression must be implemented in software.

For example, the BPF string `tcp port 80` is translated into the following abstract syntax tree:

```
[LINK IP] AND (
  [PROTO TCP] AND (
    [PORT SRC 80] OR [PORT DST 80]
```

```
)
)
```

However, this tree is still not suitable for generating configuration for monitoring cards. Packet matching in these cards is usually based on CAMs, either real hardware CAM devices or firmware implementation of the CAM function.

A CAM consists of set of rows that are all matched in parallel to parts of the incoming packet in parallel. Parts to match are defined by a mask associated with each row. The output is the row number that matched the input (usually the first such row if more rows matched). CAMs can be seen as a logical disjunction of rows and the parts of rows to match as a logical conjunction. This is exactly a property of DNF expressions, which can therefore be easily written into CAMs.

The above abstract syntax tree is not in the DNF due to the nested OR within an AND conjunction. Transformation into the DNF in the optimization process uses distributive law [11] to eliminate inner disjunctions.

Transformed abstract syntax tree follows:

```
(([LINK IP] AND [PROTO TCP]
  AND [PORT SRC 80])
OR
([LINK IP] AND [PROTO TCP]
  AND [PORT DST 80]))
```

3.3: Supporting different hardware and software environments

We will illustrate example implementation for two common monitoring cards, DAG cards and COMBO cards. DAG cards are available in various types for Ethernet, PoS and ATM links up to 10 Gb/s. COMBO cards are available for 1 Gb/s and 10 Gb/s Ethernet. Filtering and classification requirements that can be hardware-accelerated with the DAG and COMBO cards are summarized in Table 3.3. If such specification is used in a monitoring application, it will be hardware-accelerated. Those filtering or classification requirements that do not satisfy these conditions will be returned by the translation process as not fitting into resources of a monitoring card and will be implemented in software of MAPI middleware transparently to the application.

When a DAG card with DSM classification is used, the translation process will convert the BPF string `TCP and port 80` into the following specification for the `dsm_loader` utility, which will configure the DSM unit in the DAG card:

```
<filter>
<name>filter0</name>
<number>0</number>
<ethernet>
  <ipv4>
    <tcp>
      <source-port>
        <port>80</port>
        <mask hex="true">FFFF</mask>
      </source-port>
    </tcp>
  </ipv4>
</ethernet>
```

Card type	Resources
DAG (cards with coprocessor)	One filter condition based on common fields in IP, TCP and UDP headers.
DAG (all cards with DSM [12] support)	Up to 8 independent classification conditions based on comparison of arbitrary bytes in the first 64 bytes of an Ethernet frame. When packets need to be passed to the host PC for further software processing, one of those 8 conditions needs to be used to filter packets to be passed packets to the PC.
COMBO (with NIFIC [13] firmware)	8 selected 32-bit words in L2-L4 packet headers are compared to an 8192-line CAM then a sequence of arithmetic comparisons can be executed depending on the required speed.

Table 2. Resources for hardware acceleration

```

</filter>
<filter>
<name>filter1</name>
<number>1</number>
<ethernet>
  <ipv4>
    <tcp>
      <dest-port>
        <port>80</port>
        <mask hex="true">FFFF</mask>
      </dest-port>
    </tcp>
  </ipv4>
</ethernet>
</filter>

```

When a COMBO card is used, the translation process will compile the above BPF string into the following specification for the `lupgen` utility [6], which will create a configuration bitstream to be uploaded into the COMBO card:

```

:rules
{filter_0} @if_id=h0/h0 \
  @l3_reg=h0000/h8000 @protocol=h0006 @src_port=80;
{filter_0} @if_id=h0/h0 \
  @l3_reg=h0000/h8000 @protocol=h0006 @dst_port=80;
{default};

```

4: Evaluation

In this section we present comparison of packet classification performance of the same monitoring application for the 10 Gigabit Ethernet running over i) a regular Ethernet NIC, ii) a DAG card with software classification (DSM disabled) and iii) a DAG card with hardware classification (DSM enabled). The application used one classification condition on IP and UDP headers and counted packets that passed this condition. As a regular

Frame size	Myricom 10GE NIC		DAG without DSM		DAG with DSM	
	Throughput	CPU load	Throughput	CPU load	Throughput	CPU load
1518	8100	76.19	10000	0.06	10000	0.06
1280	6770	73.03	10000	0.10	10000	0.06
1024	5440	68.56	10000	0.13	10000	0.06
512	3050	68.93	10000	13.87	10000	0.06
256	1620	74.26	10000	33.56	10000	0.06
128	1130	75.56	10000	54.97	10000	0.06
64	710	77.82	10000	78.94	10000	0.06

Table 3. Performance of packet classification — throughput in Mb/s and CPU load in %

NIC, we used Myrinet 10GE PCI-E adapter. According to our other tests, this adapter appears to have highest performance of current 10 Gb/s NICs. The Myrinet card driver was version 1.3.0. The PC used was Supermicro X7DB8 mainboard with two dual-core 3 GHz Woodcrest Xeon processors with SuSe Linux and kernel version 2.6.19. However, different number of CPU cores were used in different tests depending on how packets could be distributed into multiple cores. With an Ethernet NIC and a DAG card with DSM classification, two cores were used (one for MAPI and one for application), while for the test with a DAG card and software classification, three cores were used (two for MAPI and one for application).

The achieved throughput and associated average CPU load (over the used CPU cores) in the host PC for various packet sizes is shown in Table 4. We measured performance for Ethernet frame sizes recommended in [7]. We can see that a regular Ethernet NIC did not achieve full line rate. With the DAG card, we can classify packets at the full line rate. For software classification this was achieved due to much more effective packet transfer from the network to the memory in the host PC. However, the CPU load was high and therefore little further packet processing would be possible. With hardware classification, the CPU was used only to retrieve packet counters from the DAG card and therefore it remained almost fully available for further packet processing. We did not have 10 Gb/s COMBO card available for testing.

5: Related work

Other common languages for filtering or classification description include FPL-3 and Netfilter. FPL-3 [8] goes beyond packet header comparison by adding payload searching and it is based on a data flow-driven model for distribution of packet processing into parallel engines specialized for particular protocols or tasks. The implementation is targeted for network processor hardware. Netfilter [9] is a language and packet filtering framework for Linux kernels that is particularly designed for firewall and NAT applications.

We decided to use BPF, because it is the most commonly used language in passive monitoring applications that are our targeted use area. BPF also covers the range of

expressions that can be implemented with the commonly available monitoring cards. We use FPGA because it can be used for a broader range of applications than network processors, which are a more specialized platform. Also, boards with high-speed networks processors are more expensive than boards with fast FPGAs.

6: Conclusion

We have proposed an architecture that allows monitoring applications to utilize as much as possible of hardware resources in various monitoring cards to accelerate packet filtering and classification. This allows applications to run transparently over various current and future hardware. Application programmers do not need to care about details of different monitoring cards (the programmers extending our environment for future monitoring cards of course need to do that). Applications can run in software only when monitoring moderate volumes of traffic, such as in wireless networks and in hardware-software environment for high-speed monitoring.

In our future work we plan to explore possibilities of hardware acceleration of more monitoring functions, such as payload searching or more complex statistics. This should be possible with some of the emerging high-speed (10 Gb/s) FPGA-based cards with open hardware interfaces, which allow users to write their own firmware.

References

- [1] DAG cards, Endace corporation, www.endace.com.
- [2] COMBO cards, Liberouter project, www.liberouter.org.
- [3] Libpcap library, <http://www.tcpdump.org>.
- [4] MAPI - Monitoring Application Programmable Interface. <http://mapi.uninett.no>.
- [5] Bison and Flex tools, <http://www.gnu.org/software/bison>, <http://www.gnu.org/software/flex>.
- [6] LUP configuration tool, Liberouter project, https://www.liberouter.org/wiki/index.php/LUP_configuration_tool.
- [7] S. Bradner, J. McQuaid. *Benchmarking Methodology for Interconnect Devices*, RFC2544, March 1999.
- [8] Mihai Cristea, Willem de Bruijn, Herbert Bos. *FPL-3: Towards Language Support for Distributed Packet Processing*, IFIP Networking'05, 2005, Waterloo, Canada.
- [9] The netfilter.org project, www.netfilter.org.
- [10] Mendelson, E. *Introduction to Mathematical Logic*, 4th ed. London, Chapman & Hall, p. 30, 1997.
- [11] Distributive law from elementary algebra <http://en.wikipedia.org/wiki/Distributivity>.
- [12] Data Stream Management API, Endace Corporation, <http://www.endace.com>.
- [13] NIFIC firmware, Liberouter project, www.liberouter.org/nific.php.

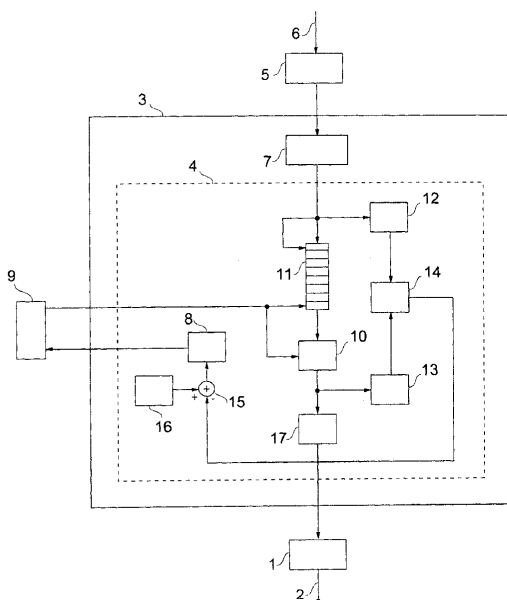
- 5.6 J. Halak, S. Ubik, and P. Zejdl. A DEVICE FOR RECEIVING OF HIGH-DEFINITION VIDEO SIGNAL WITH LOW-LATENCY TRANSMISSION OVER AN ASYNCHRONOUS PACKET NETWORK, International Patent**

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau(43) International Publication Date
29 September 2011 (29.09.2011)(10) International Publication Number
WO 2011/116735 A3

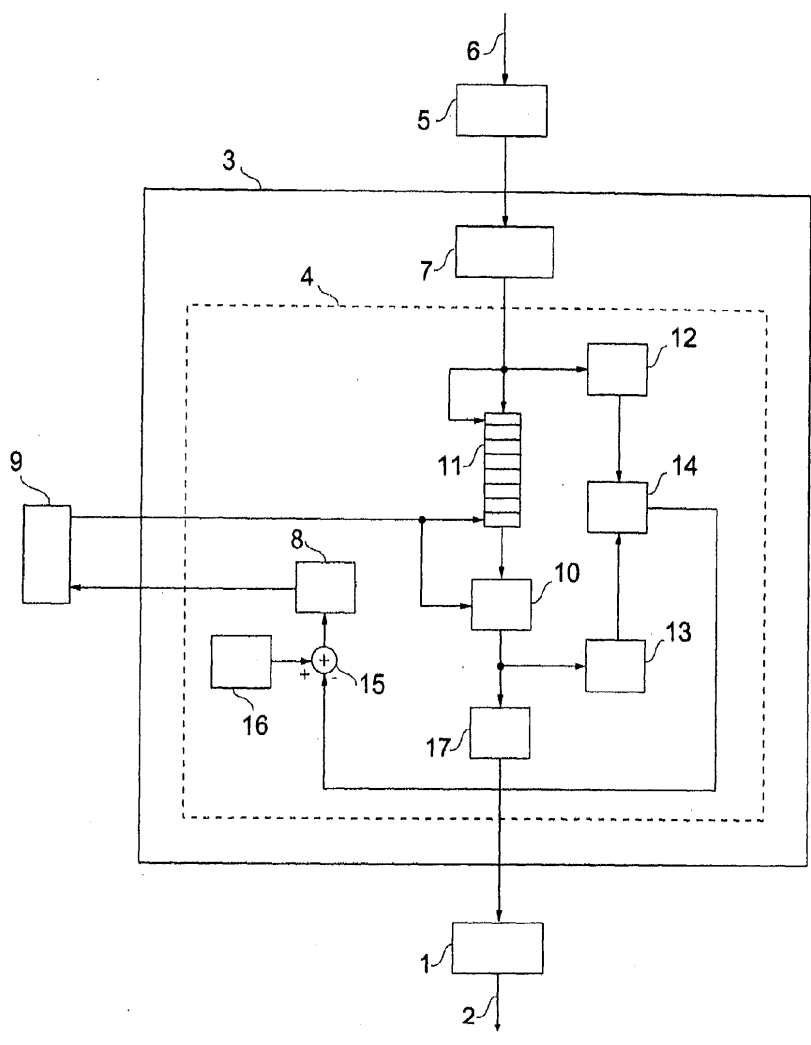
- (51) **International Patent Classification:**
H04N 21/00 (2011.01) *H04N 7/24* (2011.01)
- (21) **International Application Number:**
PCT/CZ2011/000024
- (22) **International Filing Date:**
21 March 2011 (21.03.2011)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (30) **Priority Data:**
PV 2010-226 26 March 2010 (26.03.2010) CZ
- (71) **Applicant (for all designated States except US):** CESNET, ZÁJMOVÉ SDRUŽENÍ PRAVNICKÝCH OSOB [CZ/CZ]; Zikova 4, 160 00 Praha 6 (CZ).
- (72) **Inventors; and**
- (75) **Inventors/Applicants (for US only):** HALÁK, Jiří [CZ/CZ]; Příčná 228, 29402 Kněžmost (CZ). UBIK, Sven [CZ/CZ]; Ctěnická 690, 19000 Praha 9 - Prosek (CZ). ŽEJDL, Petr [CZ/CZ]; Marie Podvalové 920/3, 19600 Praha 9 - Čakovice (CZ).
- (74) **Agent:** DUŠKOVÁ, Hana; Czech Technical University in Prague, Zikova 4, 16627 Praha 6 (CZ).
- (81) **Designated States (unless otherwise indicated, for every kind of national protection available):** AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.
- (84) **Designated States (unless otherwise indicated, for every kind of regional protection available):** ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) **Title:** A DEVICE FOR RECEIVING OF HIGH-DEFINITION VIDEO SIGNAL WITH LOW-LATENCY TRANSMISSION OVER AN ASYNCHRONOUS PACKET NETWORK

(57) **Abstract:** A device based on the proposed solution allows high-definition video transmissions with low latency over an asynchronous packet computer network such as Ethernet. The transmitter and receiver comprise a video input or output module (1), an FPGA board (3), and an optical transceiver (5) for transmission and reception of a signal over the Ethernet network. The principle of the new device is that the receiver comprises one or more tunable oscillators (9) connected to the FPGA board (3) comprising a module (7) for packet reception, and one or more sets (4) of modules for video data processing. These sets (4) of modules adjust the data rate onto video outputs (2) to the data generation rate on the transmitter's side and enable to display the beginning of a frame at the correct position, even though the transmitter and receiver are interconnected by an asynchronous packet network such as Ethernet, which cannot be used for receiver clock recovery in line with a transmission data rate, and this is all possible without the need for a large frame memory increasing the delay.

 WO 2011/116735 A3



US 2012/0327302 A1

Dec. 27, 2012

1

**DEVICE FOR RECEIVING OF
HIGH-DEFINITION VIDEO SIGNAL WITH
LOW-LATENCY TRANSMISSION OVER AN
ASYNCHRONOUS PACKET NETWORK**

FIELD OF THE INVENTION

[0001] The technical solution relates to the high-definition (HD, 2K, 4K and more) video signal transmission over a packet computer network. It belongs to the area of telecommunication technology and services.

DESCRIPTION OF THE PRIOR ART

[0002] There are several known categories of solutions for video signal transmission over a network.

[0003] The first category includes single-purpose systems converting an electrical signal from video inputs directly into an electrical or optical signal suitable for network transmission and performing the reverse conversion at the recipient side of transmission. As the video signal is not encapsulated, into frames or packets, it can only be transmitted over a dedicated link of limited Length, not over the Internet.

[0004] The second category comprises the equipment that encapsulates video data into frames for transmission over a synchronous computer network such as SONET/SDH. In this case the clock of a video receiver can be derived from the clock of a synchronous computer network. Therefore this solution is not suitable for transmission over an asynchronous computer network such as Ethernet.

[0005] The third category consists of devices that encapsulate video data into packets for transmission over an asynchronous computer network such as Ethernet. The device can be either a PC-based systems equipped with suitable cards for video signal input and output (grabber cards, video adapters, compressing cards) and a card for network transmission (network card) inserted into PC slots, or a specialized equipment. The differences in transmitter and receiver rates are solved by a sufficiently sized buffer at a receiver side which, however, as a consequence increases the delay of video transmission. In the case of PC-based systems, to transmit high-definition video signals, a complex system consisting of many cards and possibly more PCs is required limiting its portability.

[0006] The above analysis shows that a system for high-definition video transmission with low latency over an asynchronous packet computer network such as Ethernet is very difficult to obtain with existing technology.

SUMMARY OF THE INVENTION

[0007] The proposed solution of a device for high-definition video network transmission eliminates the disadvantages shown above. The transmitter and receiver comprise a video input or output module for video data input or output through one or more video inputs or outputs, an FPGA (Field-Programmable Gate Array) board, and an optical transceiver for transmission and reception of a signal over the Ethernet network. The principle of the new device is that the receiver comprises one or more tunable oscillators connected to the FPGA board comprising a module for packet reception, and one or more sets of modules for video data processing. The number of tunable oscillators equals the number of video data processing module sets. Each video data processing module set on the FPGA board in the receiver comprises a buffer, where its data input and writing clock input are connected via

the packet reception module to the electrical output of the optical transceiver, and its data output is connected to the input of a video processor. The video processor output is connected via a channel synchronization module to the video output module. The module set in addition comprises a counter whose one input is connected via the first detector of a selected row in a frame to the buffer data input, and whose another input is connected via the second detector of a selected row in a frame to the video processor output. Counter output is connected to the inverting input of a subtractor, whose positive input is connected to the memory of a required regulation value, and whose output is a difference against the required value put to the PID (Proportional-Integral-Derivative) regulator input. The PID regulator controls via its output the frequency of a tunable oscillator that is connected to the video processor clock input and buffer reading clock input.

[0008] In one embodiment, video outputs can be in the form of one or more SMPTE (Society of Motion Picture and Television Engineers) 259M (SDI, Serial Digital Interface), and/or SMPTE 292M (HD-SDI), and/or SMPTE 424 (3G-SDI), and/or SMPTE 372 (dual-link HD-SDI) channels.

[0009] In an advantageous embodiment, the PID regulator module can be implemented as a program for a processor embedded in an FPGA circuit on the FPGA board.

[0010] The proposed device is characterized by the receiver's capability to adjust the data rate onto video outputs to the data generation rate on the transmitter's side, and to display the beginning of a frame at the correct position, even though the transmitter and receiver are interconnected by an asynchronous packet network such as Ethernet, which cannot be used for receiver clock recovery in line with a transmission data rate, and this is all possible without the need for a large frame memory increasing the delay.

[0011] The instant video data volume in the FIFO-type buffer depends on delay variation when data is transmitted over a network. When the transmitter and receiver clocks differ, the buffer is systematically slowly emptied or overflowing, ultimately leading to a loss in transmitted video data. The suggested solution eliminates this problem.

BRIEF DESCRIPTION OF THE DRAWINGS

[0012] An example of a device for receiving high-definition video data with low latency over an asynchronous packet network based on the proposed solution is schematically shown in the enclosed drawing.

DETAILED DESCRIPTION OF THE PREFERRED
EMBODIMENTS:

[0013] A device for receiving high-definition video data over an asynchronous packet network can be described by the following functional blocks (see enclosed schematic diagram): video output module 1, video outputs 2, FPGA board 3, packet reception module 7, one or more video data processing module sets 4, optical transceiver 5, Ethernet interface 6, one or more tunable oscillators 9. Each video data processing module set 4 comprises: PID regulator 8, video processor 10, buffer 11, first detector 12 of a selected row in a frame, second detector 13 of a selected row in a frame, counter 14, subtractor 15, memory 16 of required regulation value, and channel synchronization module 17.

[0014] An electrical output of an optical transceiver 5 is connected to the input of the FPGA board 3, whose output is connected to the input of the video output module 1 leading to

US 2012/0327302 A1

Dec. 27, 2012

2

video outputs 2. A control input of each tunable oscillator 9 and its frequency output are connected across the FPGA board 3 to the related video data processing module set 4. Data input and writing clock input of a buffer 11 are connected across a packet reception module 7 to an electrical output of the optical transceiver 5. Data output of the buffer 11 is connected to the input of the video processor 10, the output of which leads across a channel synchronization module 17 to the video output module 1. The first input of the counter 14 is across the first detector 12 of a selected row in a frame interconnected to data input of buffer 11, and the second input of the counter 14 is across the second detector 13 of a selected row in a frame interconnected to the output of the video processor 10. The output of the counter 14 is interconnected with an inverting input of the subtractor 15, whose positive input connects to the memory 16 of required regulation value, and whose output is connected to the input of the PID regulator 8. The output of the PID regulator 8 is connected to the control input of a tunable oscillator 9, whose frequency output is connected to the clock input of the video processor 10 and to the reading clock input of the buffer 11.

[0015] An optical transceiver 5 converts the signal between the Ethernet interface 6 and its electrical output. A packet reception module 7 decapsulates video data from packets incoming from an Ethernet network, which means that it typically deploys communications protocols at link, network, and transport network layers. A packet reception module 7 further splits video data into individual module sets 4 for video data processing according to the video outputs for which it is determined. A video processor 10 converts video data into video output formats. A channel synchronization module 17 synchronizes video output groups by framing. A video output module 1 performs voltage and impedance adjustments between the FPGA board 3 and video outputs 2.

[0016] A counter 14 can be initialized by a row of the selected number coming into buffer 11 according to the data from the first detector 12 of a selected row in a frame and it can be stopped by a row of this number outgoing from the video processor 10 according to the data from the second detector 13 of a selected row in a frame. In that case the value of the counter 14 becomes positive. Alternatively, the counter 14 can be initialized by a row of the selected number outgoing from the video processor 10 according to the data from the second detector 13 of a selected row in a frame and it can be stopped by a row of this number coming into buffer 11 according to the data from the first detector 12 of a selected row in a frame. In that case the value of the counter 14 becomes negative. If the row output into video outputs 2 advances its entry into buffer 11, the video processor 10 sends an alternative row to video outputs 2, for example a copy of the preceding row.

[0017] If the average delay of a selected row between the input of the buffer 11 and the output of the video processor 10, as determined by the counter 14, differs in the subtractor 15 from the content of memory 16 of the required regulation value, the PID regulator 8 changes the frequency of the tunable oscillator 9 in order to equalize the delay to the required value. Regulation can use any row in a frame, typically the first visible one.

[0018] This method of determining regulated value delta along with a selection of the required regulation value also enables a stabilization of the frame position, that is displaying the beginning of a frame in the correct position. The regulation can be deployed in any digital video data transmission in

which it is possible to determine the sequence numbers of rows within a frame. A required regulation value depends on the video output types and the frame format and has to be empirically set for highest video stability. Using a positive required regulation value enables to set up the advance between a selected row enters a buffer and when it is displayed, and so optimize filling the buffer 11.

[0019] Video outputs can comprise, for example, one or more SMPTE 259M (SDI), and/or SMPTE 292M (HD-SDI), and/or SMPTE 424 (3G-SDI), and/or SMPTE 372 (dual-link HD-SDI) channels.

[0020] Useful characteristics of dividing the video outputs 2 into groups of one or more members where each group is connected across the video output module 1 to an independent video data processing module set 4 with an independent tunable oscillator 9 is the capability of each group of video outputs 2 to be deployed for video signal transmission from an independent video source at the transmitter side, for example, independent HD or 2K video signals.

[0021] Another useful characteristic is the frame synchronization capability within each group of video outputs 2 in the channel synchronization module 17. Synchronized video outputs 2 can be used for transmission of a higher definition (4K or more) signal split into parts or for a stereoscopic transmission (3D).

[0022] In one embodiment the video outputs 2 comprise one or more SMPTE 259M (SDI), and/or SMPTE 292M (HD-SDI), and/or SMPTE 424 (3G-SDI), and/or SMPTE 372 (dual-link HD-SDI) channels. Data signals on these connectors can also contain related audio channels.

[0023] Besides the video signal, the device can also process and transmit one or more audio channels. Format of transmitted data over a network typically relates to the format of video inputs and outputs, usually excluding dark frame parts not containing any audio signal, or to recommendations for video data transmission (e.g., RFC 4175).

INDUSTRIAL APPLICABILITY

[0024] The technical solution is well suitable for industrial applications in private, local, national, and international computer networks for high-definition video signal transmissions including real-time and low-latency transmissions, for example for remote interactive access to lectures, medical surgeries or film recordings in during post-production phase and for their presentation.

1. A device for receiving of high-definition video signal with low latency over an asynchronous packet network comprising a video output module for video data output through one or more video outputs, an FPGA board, and an optical transceiver 5 for signal reception through an Ethernet interface 6, wherein the FPGA board comprises a packet reception module and one or more video data processing module sets, where an independent tunable oscillator with its control input and frequency output is connected across the FPGA board to each module set, which comprises a buffer, whose data input and writing clock input are connected across a packet reception module to the electrical output of the optical transceiver, and whose data output is connected to the input of the video processor, whose output goes across a channel synchronization module to the video output module, whereas each module set comprises a counter for determining delay of a selected row, whose first input is across the first detector of a selected row in a frame connected to the input of the buffer, and whose second input is across the second detector of a selected row in

US 2012/0327302 A1

Dec. 27, 2012

3

a frame connected to the output of the video processor, while the output of the counter is connected with an inverting input of a subtractor, whose positive input is connected to the memory of the required regulation value, where the output of the subtractor is connected to the input of the PID regulator, whose output is connected to the control input of the tunable oscillator, whose frequency output is connected to the clock input of the video processor and to the reading clock input of buffer.

2. The device according to claim 1, wherein the video outputs comprise one or more SMPTE 259M, and/or SMPTE 292M, and/or SMPTE 424, and/or SMPTE 372 channels.

3. The device according to claim 1, wherein the PID regulator module is implemented as a program for a processor embedded in an FPGA circuit on the FPGA board 3.

* * * * *

Bibliography

- [1] J. Klaue, B. Rathke, and A. Wolisz, “EvalVid - A Framework for Video Transmission and Quality Evaluation,” in *Computer Performance Evaluation. Modelling Techniques and Tools*, vol. 2794 of *Lecture Notes in Computer Science*, pp. 255–272, Springer Berlin Heidelberg, 2003.
- [2] J. Aweya, D. Y. Montuno, M. Ouellette, and K. Felske, “Clock recovery based on packet inter-arrival time averaging,” *Computer Communications, Elsevier B.V.*, vol. 29, no. 10, pp. 1696 – 1709, 2006. Monitoring and Measurements of IP Networks.
- [3] “IEEE 802.3TM Industry Connections Ethernet Bandwidth Assessment,” Technical Report, IEEE 802.3 Ethernet Working Group, July 2012. Available: http://www.ieee802.org/3/ad_hoc/bwa/BWA_Report.pdf.
- [4] CESNET Press Release, “CESNET2 national network completely ready for 100 Gbps.” Available: <http://archiv.ces.net/doc/press/2012/pr120621.html>.
- [5] “IMAX on Wikipedia.” Available: <http://en.wikipedia.org/wiki/IMAX>.
- [6] L. Renambot, A. Rao, R. Singh, B. Jeong, N. Krishnaprasad, V. Vishwanath, V. Chandrasekhar, N. Schwarz, and A. Spale, “SAGE: the Scalable Adaptive Graphics Environment,” in *Proc. 4th Workshop on Advanced Collaborative Environments (WACE 2004)*, (Nice, France), September 2004.
- [7] K. Ponto, K. Doerr, and F. Kuester, “Giga-stack: A method for visualizing giga-pixel layered imagery on massively tiled displays,” *Future Generation Computer Systems, Elsevier B.V.*, vol. 26, no. 5, pp. 693 – 700, 2010.
- [8] C. Cruz-Neira, D. J. Sandin, and T. A. DeFanti, “Surround-screen projection-based virtual reality: the design and implementation of the cave,” in *Proc. 20th annual conference on Computer graphics and interactive techniques, SIGGRAPH '93*, (New York, NY, USA), pp. 135–142, ACM, 1993.
- [9] T. A. DeFanti, G. Dawe, D. J. Sandin, J. P. Schulze, P. Otto, J. Girado, F. Kuester, L. Smarr, and R. Rao, “The StarCAVE, a third-generation CAVE and virtual reality OptIPortal,” *Future Generation Computer Systems, Elsevier B.V.*, vol. 25, no. 2, pp. 169 – 178, 2009.
- [10] CESNET, “MVTP-4K: Modular Video Transfer Platform.” Available: <http://www.cesnet.cz/research/technologies-for-network-applications/?lang=en>.
- [11] 4K Gateway, “Modular Video Transfer Platform.” Available: <http://www.4kgateway.com/>.
- [12] Intuitive Surgical, Inc., “The da Vinci Surgical System.” Available: http://www.intuitivesurgical.com/products/davinci_surgical_system/.

- [13] C. Poynton, *Digital Video and HDTV Algorithms and Interfaces*, ch. Chroma subsampling notation, p. 90. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1 ed., 2003.
- [14] T. DeFanti, C. de Laat, J. Mambretti, K. Neggers, and B. S. Arnaud, “TransLight: a global-scale LambdaGrid for e-science,” *Communications of the ACM*, vol. 46, pp. 34–41, November 2003.
- [15] J. Nielsen, *Usability Engineering*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.
- [16] ITU-T, International Telecommunication Union, “Recommendation G.114: One-way transmission time, ITU-T Study Group 12.,” May 2003.
- [17] ITU-T, International Telecommunication Union, “Recommendation G.709: Interfaces for the optical transport network (OTN).”
- [18] V. Novak, J. Verich, and J. Mer, “CESNET 2 100 GE IPoDWDM Early Field Test in the Live Network,” Technical Report, CESNET, July 2012. Available: <http://www.cesnet.cz/wp-content/uploads/2013/01/100ge-test.pdf>.
- [19] SMPTE, “Society of Motion Picture and Television Engineers.” Available: <https://www.smpte.org/>.
- [20] SMPTE Standard, “SMPTE-259M: for Television - SDTV Digital Signal/Data - Serial Digital Interface.”
- [21] SMPTE Standard, “SMPTE-292M: 1.5 Gb/s Signal/Data Serial Interface.”
- [22] SMPTE Standard, “SMPTE-428-0: D-Cinema Distribution Master - Image Pixel Structure Level 3 - Serial Digital Interface Signal Formatting, SMPTE 428-9-2008..”
- [23] SMPTE Standard, “SMPTE-372M: Dual Link 1.5 Gb/s Digital Interface for 1920 x 1080 and 2048 x 1080 Picture Formats, SMPTE 372-2009, Society of Motion Picture and Television Engineers..”
- [24] SMPTE Standard, “SMPTE-424M: for Television - 3 Gb/s Signal/Data Serial Interface.”
- [25] Blackmagic Design, “6G-SDI, Blackmagic Design Announces Worlds First 6G-SDI Products, Press Release.” Available: <http://www.blackmagicdesign.com/press/pressdetails?releaseID=38169>.
- [26] RFC3550, “RTP: A Transport Protocol for Real-Time Applications <http://tools.ietf.org/html/rfc3550>.” Available: <http://tools.ietf.org/html/rfc3550>.
- [27] RFC4960, “Stream Control Transmission Protocol.” Available: <http://tools.ietf.org/html/rfc4960>.

- [28] S. Akhshabi, A. C. Begen, and C. Dovrolis, “An experimental evaluation of rate-adaptation algorithms in adaptive streaming over HTTP,” in *Proc. Second annual ACM conference on Multimedia systems, MMSys '11*, (New York, NY, USA), pp. 157–168, ACM, 2011.
- [29] ISO Standard, “ISO/IEC 14496: Information technology - Coding of audio-visual objects,” 2001. (MPEG-4).
- [30] C. Sreenan, C. Jyh-Cheng, P. Agrawal, and B. Narendran, “Delay reduction techniques for playout buffering,” *IEEE Transactions on Multimedia*, vol. 2, no. 2, pp. 88–100, 2000.
- [31] S. B. Moon, J. Kurose, and D. Towsley, “Packet audio playout delay adjustment: performance bounds and algorithms,” *Multimedia Systems*, vol. 6, no. 1, pp. 17–28, 1998. ACM/Springer-Verlag.
- [32] P. DeLeon and C. Sreenan, “An adaptive predictor for media playout buffering,” in *Proc. 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 6, pp. 3097–3100, IEEE Computer Society, 1999.
- [33] RFC793, “Transmission control protocol specification,” September 1981. ARPANET Working Group Request For Comment. Available: <http://www.ietf.org/rfc/rfc793.txt>.
- [34] V. Jacobson, “Congestion avoidance and control,” *ACM SIGCOMM Computer Communication Review*, vol. 18, pp. 314–329, August 1988.
- [35] R. Ramjee, J. Kurose, D. Towsley, and H. Schulzrinne, “Adaptive playout mechanisms for packetized audio applications in wide-area networks,” in *Proc. 13th IEEE Conference on Networking for Global Communications, INFOCOM '94*, pp. 680–688 vol.2, 1994.
- [36] H. Marjamaki and R. Kantola, “Performance evaluation of an ip voice terminal,” in *Proc. Fifth IFIP Conference on Intelligence in Networks*, vol. 32 of *IFIP Advances in Information and Communication Technology*, pp. 349–362, Springer US, 2000.
- [37] D. L. Stone and K. Jeffay, “An empirical study of delay jitter management policies,” *Multimedia Systems*, vol. 2, pp. 267–279, January 1995. ACM/Springer-Verlag.
- [38] R. C. Lau and P. E. Fleischer, “Synchronous techniques for timing recovery in BISDN,” *IEEE Transactions on Communications*, vol. 43, no. 234, pp. 1810–1818, 1995.
- [39] R. P. Singh and S. H. Lee, “Adaptive clock synchronization schemes for real-time traffic in broadband packet networks,” in *Proc. 8th European Conference on Electrotechnics, EUROCON 88*, pp. 84–88, 1988.

- [40] R. P. Singh, S.-H. Lee, and C.-K. Kim, "Jitter and clock recovery for periodic traffic in broadband packet networks," in *Proc. 1988 IEEE Global Telecommunications Conference and Exhibition on 'Communications for the Information Age.'*, GLOBECOM '88. IEEE Transactions on Communications, pp. 1468–1473 vol.3, 1988.
- [41] K. S. Kim and B. G. Lee, "KALP: a Kalman filter-based adaptive clock method with low-pass prefiltering for packet networks use," *IEEE Transactions on Communications*, vol. 48, no. 7, pp. 1217–1225, 2000.
- [42] ITU-T, International Telecommunication Union, "Recommendation Y.IMPtdmpls: Implementors Guide for TDM-MPLS network interworking, ITU-T Study Group 13.," 2001-2004.
- [43] D. Shirai, T. Kawano, T. Fujii, K. Kaneko, N. Ohta, S. Ono, S. Arai, and T. Ogoshi, "Real time switching and streaming transmission of uncompressed 4K motion pictures," *Future Generation Computer Systems, Elsevier B.V*, vol. 25, no. 2, pp. 192 – 197, 2009.
- [44] T. Shimizu, D. Shirai, H. Takahashi, T. Murooka, K. Obana, Y. Tonomura, T. Inoue, T. Yamaguchi, T. Fujii, N. Ohta, S. Ono, T. Aoyama, L. Herr, N. van Osdol, X. Wang, M. D. Brown, T. A. DeFanti, R. Feld, J. Balsler, S. Morris, T. Henthorn, G. Dawe, P. Otto, and L. Smarr, "International real-time streaming of 4K digital cinema," *Future Generation Computer Systems, Elsevier B.V*, vol. 22, no. 8, pp. 929 – 939, 2006.
- [45] D. Shirai, M. Kitamura, T. Fujii, A. Takahara, K. Kaneko, and N. Ohta, "Multi-point 4K/2K layered video streaming for remote collaboration," *Future Generation Computer Systems, Elsevier B.V*, vol. 27, no. 7, pp. 986 – 990, 2011.
- [46] UltraGrid, "High-quality low-latency video and audio transmissions, Laboratory of Advanced Networking Technologies (SITOLA)..” Available: <https://www.sitola.cz/igrid/index.php/About>.
- [47] P. Holub, L. Matyska, M. Liska, L. Hejtmanek, J. Denemark, T. Rebok, A. Hutanu, R. Paruchuri, J. Radil, and E. Hladka, "High-definition multimedia for multiparty low-latency interactive communication," *Future Generation Computer Systems, Elsevier B.V*, vol. 22, pp. 856 – 861, October 2006.
- [48] LOLA, "LOW LATency audio visual streaming system, Conservatorio di Musica Giuseppe Tartini from Trieste and the Italian Research and Academic Network.." Available: <http://www.conservatorio.trieste.it/artistica/lola-project/lola-low-latency-audio-visual-streaming-system>.
- [49] LOLA, "GEANT and LOLA - enabling real-time musical collaboration, Conservatorio di Musica Giuseppe Tartini from Trieste and the Italian Research and Academic Network.." Available: <http://geant3.archive.geant.net/Users/ArtsandCulture/Pages/LOLA.aspx>.

- [50] LOLA, “(Low Latency) Project Case Study: Enabling remote real time musical performances over advanced networks, Conservatorio di Musica Giuseppe Tartini from Trieste and the Italian Research and Academic Network.” Available: http://www.conservatorio.trieste.it/artistica/ricerca/progetto-lola-low-latency/lola-case-study.pdf?ref_uid=e98cac4a9c6a546ac9adebc9dea14f7b.
- [51] IntoPIX, “JPEG 2000 4K Streaming System.” Available: <http://www.intopix.com/products/index/index/id/18/lang/en>.
- [52] Net Insight, “Nimbra 600 Series of Media Switch Routers.” Available: <http://www.netinsight.net/Products/Nimbra-600-Series/>.
- [53] CineGrid, “An interdisciplinary community that is focused on the research, development, and demonstration of networked collaborative tools.” Available: <http://www.cinegrid.org/>.
- [54] FPGA, “Field Programmable Gate Array.” Available: <http://www.xilinx.com/training/fpga/fpga-field-programmable-gate-array.htm/>.
- [55] Xilinx User Guide, “UG702: Partial Reconfiguration User Guide, Xilinx Inc.” Available: http://www.xilinx.com/support/documentation/sw_manuals/xilinx14_5/ug702.pdf.
- [56] J. Halak and S. Ubik, “MTPP - Modular Traffic Processing Platform,” in *Proc. 12th IEEE Symposium on Design and Diagnostics of Electronic Systems DDECS '09*, (Liberec, Czech Republic), pp. 170–173, 2009.
- [57] J. Halak, “Multigigabit network traffic processing,” in *Proc. The International Conference on Field Programmable Logic and Applications FPL 2009*, (Prague, Czech Republic), pp. 521–524, 2009.
- [58] ITU-T, International Telecommunication Union, “Recommendation O.150: Digital test patterns for performance measurements on digital transmission equipment,” 1996.
- [59] CESNET, “Czech Academic Network Operator, Research Projects.” Available: <http://www.cesnet.cz/research/?lang=en>.
- [60] SMPTE Standard, “SMPTE-274M: for Television - 1920 x 1080 Image Sample Structure, Digital Representation and Digital Timing Reference Sequences for Multiple Picture Rates.”
- [61] Xilinx Datasheet, “DS080: System ACE CompactFlash Solution.” Available: www.xilinx.com/support/documentation/data_sheets/ds080.pdf.

- [62] V. Lai and O. Diessel, "ICAP-I: A reusable interface for the internal reconfiguration of Xilinx FPGAs," in *Proc. The International Conference on Field Programmable Logic and Applications FPL 2009*, (Prague, Czech Republic), pp. 357–360, 2009.
- [63] LOBSTER Project, "Large-scale Monitoring of Broadband Internet Infrastructures.." Available: <http://www.ist-lobster.org>.
- [64] M. Polychronakis, E. Markatos, K. Anagnostakis, and A. Oslebo, "Design of an application programming interface for ip network monitoring," in *Proc. Network Operations and Management Symposium, NOMS 2004. IEEE/IFIP*, vol. 1, (Seoul, South Korea), pp. 483–496 Vol.1, 2004.
- [65] Liberouter Project, "COMBO Monitoring Cards.." Available: <http://www.liberouter.org>.
- [66] Endace, "DAG Monitoring Cards.." Available: <http://www.endace.com/>.
- [67] MAPI, "Monitoring Application Programmable Interface.." Available: <https://trac.uninett.no/mapi>.
- [68] SCAMPI Project, "A Scaleable Monitoring Platform for the Internet.." Available: <http://www.ist-scampi.org>.
- [69] Jiri Navratil and Jan Schraml, "3D Full HD transmission of robotic-assisted surgery from Czech Republic to the APAN meeting in HongKong, February 22, 2011." Available: <http://www.apan.net/meetings/HongKong2011/Session/agendaFiles/Joint.pdf>.
- [70] CESNET Press Release, "3D Full HD transmission of robotic-assisted surgery in *Proc. 5th International Congress on Mini-Invasive and Robotic Surgery*, Brno, Czech Republic, October 19, 2010.." Available: <http://archiv.ces.net/doc/press/2010/pr101013.html>.
- [71] CESNET Press Release, "3D Full HD transmission of robotic-assisted surgery from Czech Republic to the KEK in Japan, November 23, 2010.." Available: <http://archiv.ces.net/doc/press/2010/pr101123.html>.
- [72] CESNET Press Release, "3D Full HD transmission of robotic-assisted surgery at ITU Telecom World in Geneva, October 24-27, 2011.." Available: <http://archiv.ces.net/doc/press/2011/pr111031.html>.
- [73] IIM, "Institute of Intermedia." Available: <https://www.iim.cz/?id=1&lang=1/>.
- [74] CESNET Press Release, "CESNET demonstrates the application of high-resolution video transmissions for remote collaboration at CineGrid 2011 in San Diego, December, 2011." Available: <http://archiv.ces.net/doc/press/2011/pr111229.html>.

Publications of the Author

List of Glossed Papers

- [A1] J. Halak, S. Ubik, and P. Zejdl, “Receiver synchronization in video streaming with short latency over asynchronous networks,” in *Proc. 13th IEEE Symposium on Design and Diagnostics of Electronic Circuits and Systems*, (Vienna, Austria), pp. 403–405, IEEE Computer Society, April 2010.

The paper has been cited in:

- S. Lim, J. Lee, J. Lee, and B. Goo, “Introducing experimental stereoscopic into networked live performance with very limited system resources,” in *Proc. 32nd Meeting of the Asia-Pacific Advanced Network (APAN) 2011*, vol. 32, (New Delhi, India), pp. 185–192, APAN, August 2011.

- [A2] J. Halak, M. Krsek, S. Ubik, P. Zejdl, and F. Nevrela, “Real-time long-distance transfer of uncompressed 4K video for remote collaboration,” *Future Generation Computer Systems, Elsevier B.V.*, vol. 27, no. 7, pp. 886–892, 2011.

The paper has been cited in:

- P. Holub, M. Srom, M. Pulec, J. Matela, and M. Jirman, “GPU-accelerated DXT and JPEG compression schemes for low-latency network transmissions of HD, 2K, and 4K video,” *Future Generation Computer Systems, Elsevier B.V.*, vol. 29, no. 8, pp. 1991 – 2006, 2013.
- A. O. Ejeya, and S. D. Walker, “Uncompressed quad-1080p wireless video streaming,” *4th Computer Science and Electronic Engineering Conference (CEECE)*, IEEE Xplore, University Of Essex, Colchester, pp. 13-16, September 2012.
- S. Lim, J. Lee, J. Lee, and B. Goo, “Introducing experimental stereoscopic into networked live performance with very limited system resources,” in *Proc. 32nd Meeting of the Asia-Pacific Advanced Network (APAN) 2011*, vol. 32, (New Delhi, India), pp. 185–192, APAN, August 2011.

- [A3] J. Halak, S. Ubik, and P. Zejdl, “Scalable embedded architecture for high-speed video transmissions and processing,” in *Proc. Sixth International Conference on Systems and Networks Communications, ICSNC 2011*, (Barcelona, Spain), pp. 161–166, XPS (Xpert Publishing Services), October 2011.
- [A4] S. Ubik, J. Navratil, P. Zejdl, and J. Halak, “Real-Time Stereoscopic Streaming of Medical Surgeries for Collaborative eLearning,” in *Proc. 9th International Conference on Cooperative Design, Visualization, and Engineering, CDVE 2012*, vol. 7467 of *Lecture Notes in Computer Science*, (Osaka, Japan), pp. 73–77, Springer Berlin Heidelberg, September 2012.
- [A5] P. Zejdl, S. Ubik, V. Macek, and A. Oslebo, “Traffic classification for portable applications with hardware support,” in *Proc. International Workshop on Intelligent Solutions in Embedded Systems, WISES 2008*, (Regensburg, Germany), pp. 1–9, IEEE Computer Society, July 2008.

The paper has been cited in:

- M. Zhanikeev and Y. Tanaka, “Modelling network performance of end hosts,” *IEICE Transactions on Information and Systems, The Institute of Electronics, Information and Communication Engineers*, vol. E95.D, pp. 1872–1881, July 2012.

Relevant Refereed Publications

- [A6] S. Ubik, Z. Travnicek, P. Zejdl, and J. Halak, “Remote Access to 3D Models for Research, Engineering, and Art,” *IEEE Transactions on Multimedia*, vol. 19, pp. 12–19, October 2012.
- [A7] A. Friedl, J. Halak, M. Krsek, S. Ubik, and P. Zejdl, “Low-Latency Transmissions for Remote Collaboration in Post-Production,” in *Proc. 2012 SMPTE Annual Technical Conference & Exposition*, no. 10, (Hollywood, California), pp. 1–10, SMPTE, Society of Motion Picture & Television Engineers, October 2012.
- [A8] J. Navratil, S. Ubik, P. Peciva, J. Halak, P. Zejdl, and J. Schraml, “Live video transmission as tool in teaching students and exchanging the experiences in medical fields,” in *Proc. 1st Internet & Business Conference (IBC) 2012*, (Rovinj, Croatia), University of Zagreb, June 2012.
- [A9] J. Navratil, M. Sarek, S. Ubik, J. Halak, P. Zejdl, P. Peciva, and J. Schraml, “Real-time stereoscopic streaming of robotic surgeries,” in *Proc. 13th IEEE International Conference on e-Health Networking Applications and Services, IEEE HEALTHCOM 2011*, (Columbia, Missouri, USA), pp. 40–45, IEEE Communications Society, June 2011.

- [A10] S. Ubik, P. Zejdl, J. Halak, M. Sarek, J. Schraml, and P. Peciva, “Multi-channel streaming for medical and multimedia industry application,” in *Proc. Terena Networking Conference, TNC’11*, (Prague, Czech Republic), May 2011.
- [A11] J. Halak, J. Navratil, P. Peciva, M. Sarek, S. Ubik, and P. Zejdl, “Live long-distance stereoscopic transmissions of surgical procedures,” in *Proc. 8th International Conference on Emerging eLearning Technologies and Applications, ICETA 2010*, (The High Tatras, Slovakia), pp. 115–119, elfa, October 2010.

Remaining Refereed Publications

- [A12] J. Halak, S. Ubik, and P. Zejdl, “Data stream processing for 40 Gb/s networks,” in *Proc. Fifth International Conference on Digital Telecommunications, ICDT 2010*, (Athens, Greece), pp. 149–152, IEEE Computer Society, June 2010.
- [A13] P. Zejdl and S. Ubik, “Dynamická rekonfigurace v monitorovani vysokorychlostnich siti,” *Professional Computing*, vol. 4, 2009.
- [A14] S. Ubik and P. Zejdl, “Passive monitoring of 10 Gb/s lines with PC hardware,” in *Proc. Terena Networking Conference, TNC’08*, (Bruges, Belgium), May 2008.

The paper has been cited in:

- J. Wolfgang, S. Tafvelin, and T. Olovsson, “Review: Passive internet measurement: Overview and guidelines based on experiences,” *Computer Communications, Elsevier B.V.*, vol. 33, pp. 533–550, Mar. 2010.
 - J. Novotny, P. Celeda, T. Dedek, and R. Krejci, “Hardware Acceleration for Cyber Security,” In. *IST-091 - Information Assurance and Cyber Defence*, pp. 86–101, Tallinn, Estonia, November 2010, NATO Research and Technology Organization.
- [A15] S. Ubik and P. Zejdl, “Pasivni monitorovani linek 10 Gb/s pocitacem typu PC,” *Sdelovaci technika*, vol. 6, pp. 8–10, 2008.

Unrefereed Publications

- [A16] S. Ubik and P. Zejdl, “Evaluating application–layer classification using a machine learning technique over different high speed networks,” in *Proc. Fifth International Conference on Systems and Networks Communications, ICSNC 2010*, (Nice, France), IEEE Computer Society, August 2010.

The paper has been cited in:

- T. Bujlow, K. Balachandran, T. Riaz, and J. M. Pedersen, “Volunteer-based system for classification of traffic in computer networks,” In *Proc. 19th Telecommunications Forum, TELFOR 2011*, pages 210-213, Belgrade, Serbia, November 2011. IEEE Communications Society.
 - T. Bujlow, T. Riaz, and J. M. Pedersen, “A method for classification of network traffic based on C5.0 Machine Learning Algorithm,” In *Proc. 2012 International Conference on Computing, Networking and Communications, ICNC12*, pages 237-241, Maui, Hawaii, February 2012. IEEE Communications Society.
- [A17] P. Zejdl and S. Ubik, “Lightweight application-layer classification,” in *Deliverable DJ2.3.1: Specification of Advanced Features for a Multi-Domain Monitoring Infrastructure*, GEANT GN3 Project: Multidomain Network Services Research, pp. 39–43, 2010. Available: http://www.geant.net/Media_Centre/Media_Library/Media%20Library/GN3-10-002-DJ2-3-1_Specification_of_Advanced_Features_for_a_Multi-Domain_Monitoring_Infrastructure.pdf.
- [A18] S. Ubik, P. Zejdl, and J. Halak, “Real-time anonymization in passive network monitoring,” in *Proc. Third International Conference on Networking and Services, ICNS '07*, (Athens, Greece), p. 100, IEEE Computer Society, June 2007.
- The paper has been cited in:
- A. Blake and R. Nelson, “Scalable architecture for prefix preserving anonymization of ip addresses,” in *Embedded Computer Systems: Architectures, Modeling, and Simulation*, vol. 5114 of *Lecture Notes in Computer Science*, pp. 33–42, Samos, Greece: Springer Berlin Heidelberg, 2008.
 - A. Jain, K. P. G. Kutty, M. John, and P. Balan, “Analyzing and Recommending DPI Countermeasure Tools for Protecting User Privacy”, *University of Colorado - Boulder*, April 2009.
- [A19] P. Zejdl and J. Halak, “FPGA-based acceleration of packet header anonymization,” in *Proc. 11th International Student Conference on Electrical Engineering, POSTER 2007*, (Prague), p. IC21, Czech Technical University, May 2007.
- [A20] S. Ubik, P. Zejdl, and J. Halak, “FPGA-based Packet header Anonymisation,” Tech. Rep. 16/2006, CESNET, Prague, Czech Republic, 2007. Available: <http://www.cesnet.cz/sdruzeni/dokumenty/networking-studies/networking-studies-2007/>.
- [A21] P. Zejdl, “Hardware-Supported Packet Header Anonymization,” Master’s thesis, Czech Technical University in Prague, Czech Republic, 2006. Available: <https://dip.felk.cvut.cz/browse/details.php?f=F3&d=K13136&y=2006&a=zejdlp1&t=dip1>.

Patents and Utility Patents

- [A22] S. Ubik, J. Halak, and P. Zejdl, “Device for reception of video signals transmitted through packet computer network . *Utility patent no. 25181/2012-27260.*,” 2012.
- [A23] J. Halak, S. Ubik, and P. Zejdl, “A device for receiving of high-definition video signal with low-latency transmission over an asynchronous packet network. ***International Patent no. WO 2011116735 A2.***,” 2011.
- [A24] J. Halak, S. Ubik, and P. Zejdl, “Apparatus for receiving video signal of high resolution transmitted with a small delay through asynchronous packet computer network. ***Patent no. 302423/2010-226.***,” 2010.
- [A25] J. Halak, S. Ubik, and P. Zejdl, “Modular programmable platform for high-speed hardware processing of packets. ***Patent no. 300812/2007-850.***,” 2007.
- [A26] J. Halak, S. Ubik, and P. Zejdl, “Device for reception of video signal of high resolution transmitted with small delay through asynchronous packet computer network. *Utility patent no. 20878/2010-22484.*,” 2010.
- [A27] J. Halak, S. Ubik, and P. Zejdl, “Device for processing and transmission of high resolution video signal by computer network. *Utility patent no. 20503/2009-22001.*,” 2009.
- [A28] J. Halak, S. Ubik, and P. Zejdl, “Modular programmable platform for high-speed hardware processing of packets. *Utility patent no. 18271/2007-19462.*,” 2007.

Prototypes

- [A29] J. Halak, S. Ubik, and P. Zejdl, “MVTP-4K v.2011.”
- [A30] J. Halak, S. Ubik, and P. Zejdl, “MVTP-4K 2010.”
- [A31] J. Halak, S. Ubik, and P. Zejdl, “MTPP-40 2010.”
- [A32] S. Ubik, J. Halak, and P. Zejdl, “MTPP-10.”

Appendix A

Prototypes

A.1 MTPP-10



Figure A.1: 10 Gb/s Modular Traffic Processing Platform (MTPP-10) prototype

A.2 MTPP-40



Figure A.2: 40 Gb/s Modular Traffic Processing Platform (MTPP-40) prototype

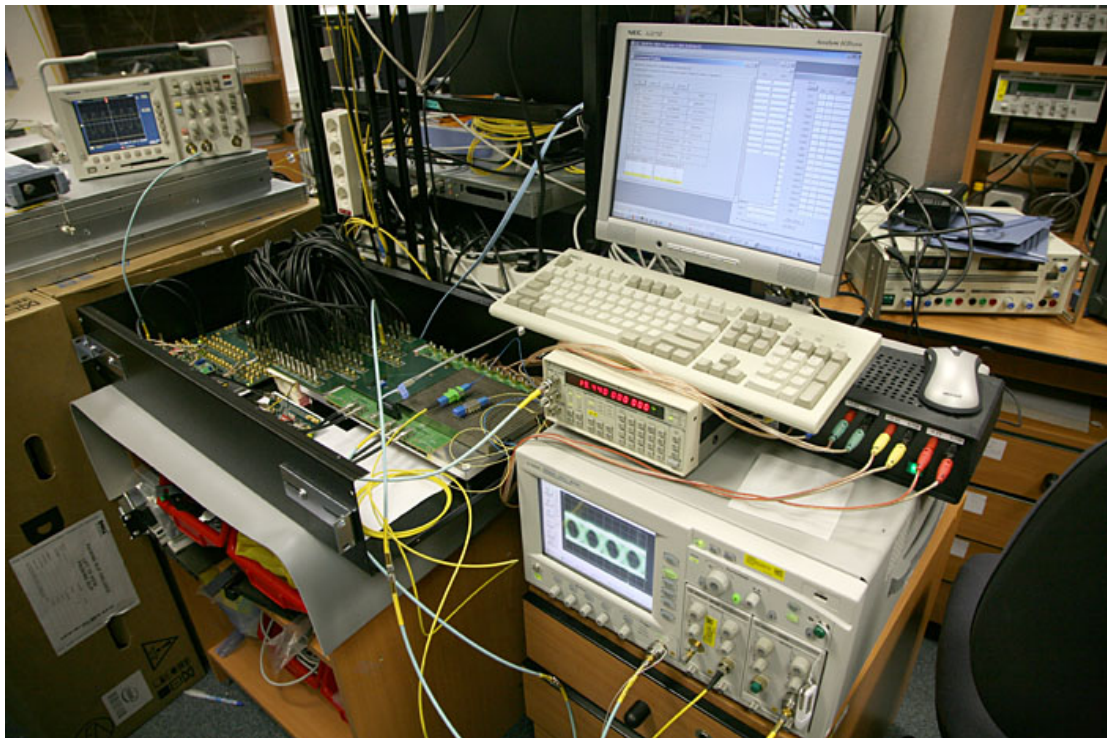


Figure A.3: MTPP-40 Optical Loopback Test Setup

A.3 MVTP-4K



Figure A.4: The first Modular Video Transfer Platform (MVTP-4K) prototype

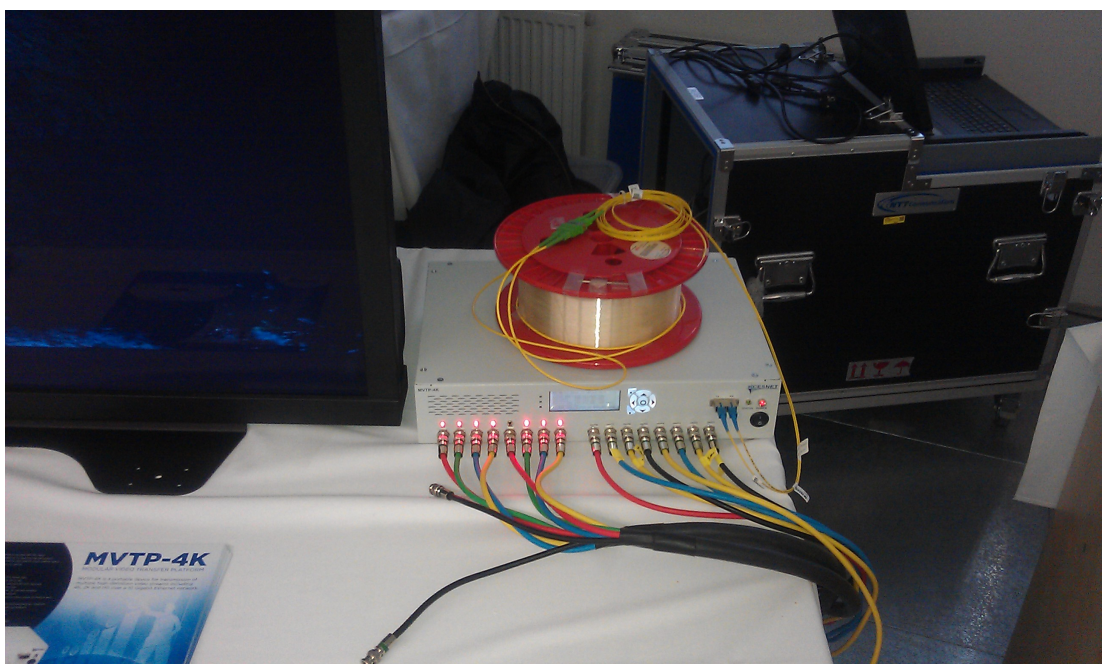


Figure A.5: MVTP-4K Optical Loopback Test Setup with 4K LCD Monitor (Astro Design DM-3400)

Appendix B

List of abbreviations

2K	Image resolution of 4096 or 3840 pixels x 2160 lines
4K	Image resolution of 2048 or 1920 pixels x 1080 lines
BPF	Berkeley Packet Filters
CAVE	Cave Automatic Virtual Environment
CBR	Constant Bitrate
FIFO	First In, First Out buffer
FPGA	Field Programmable Gate Array
Gb/s	Gigabit per second
HD-SDI	High-Definition Serial Digital Interface
JTAG	Joint Test Action Group / Standard test access and boundary-scan port
MAPI	Monitoring API
MTPP	Modular Traffic Processing Platform
MVTP	Modular Video Processing Platform
PCM	Pulse-code modulation
PID	Proportional-Integral-Derivative
PLL	Phase-Locked Loop
SAGE	Scalable Adaptive Graphics Environment
SMPTE	Society of Motion Picture and Television Engineers
VBR	Variable Bitrate