UNIVERSITY OF MACEDONIA
POSTGRADUATE PROGRAM STUDIES
DEPARTMENT OF APPLIED INFORMATICS

HAPTIC DATA TRANSFERRING THROUGH CONVERGING NETWORKS

Doctoral Thesis

of

Georgios Kokkonis

Thessaloniki, 2016

Doctoral Thesis Committee:

Manos Roumeliotis, Professor, University of Macedonia (Advisor)
Kostas Psannis, Assistant Professor, University of Macedonia (Advisor)
Stylianos Xinogalos, Assistant Professor, University of Macedonia

HAPTIC DATA TRANSFERRING THROUGH CONVERGING NETWORKS

George Kokkonis

Msc, Electrical & Computer Engineer, AUTH, 2003

Thesis submitted

in partial fulfillment of the requirements for the
Doctorate degree in Computer Science

Examination Committee:

Manos Roumeliotis, Professor, University of Macedonia

Konstantinos Margaritis, Professor, University of Macedonia

Georgios Papadimitriou, Professor, Aristotle University of Thessaloniki

Fotini – Niovi Pavlidou, Professor, Aristotle University of Thessaloniki

Vassilis Tsaoussidis, Professor, Democritus University of Thrace

Kostas Psannis, Assistant Professor, University of Macedonia

Stylianos Xinogalos, Assistant Professor, University of Macedonia

Author: Georgios Kokkonis

Thessaloniki, 2016

Abstract

As the Internet spreads, new web applications come to light. One promising sector that is still in its infancy is supermedia applications. Supermedia applications manipulate video, audio, haptic and other sensory data. With the word haptic we refer to the sense of touch that the user feels when he uses a “Haptic” service. The haptic feeling has the ability to increase the sense of reality, to excite the user and improve the quality of experience. To carry out this sense through the Internet was, until recently, impracticable due to processing inefficiencies and/or protocol performance in capabilities, such as throughput and jitter constraints. This thesis presents a survey of transport protocols for supermedia applications. It outlines the Haptic data transmission characteristics and the necessary QoS requirements for the maximization of the Quality of Experience for Haptic users. It also depicts the qualitative features that transport and application layer protocols should contain in order to carry haptic data.

It also describes a Haptic system architecture. A new network adaptive flow control algorithm is proposed. The new algorithm combines most of the known flow control algorithms while taking into account the network conditions of the Internet and the significant haptic events.

It analyses the metrics that have to be taken into consideration for the evaluation of Haptic transferring. These metrics are the delay, the jitter, the throughput, the efficiency, the packet loss and the proposed by the authors, packet arrival deviation. Based on these metrics, evaluation of the most commonly used real time transport protocols is performed.

It also presents experiments for real time Haptic data transferring that have been carried out by the authors through different networks and locations. Extensive simulations and experiments for the performance evaluation of transport protocols for real time transferring HEVC streams with supermedia data are carried out. Complements, differences and relevancies between simulation and real world experiments are discussed. The simulation tests reveal which protocols could be used for the transfer of real-time supermedia data with a HEVC video stream.

As far as video transmission is concerned, this thesis presents the related work on High Efficiency Video Coding. It points out the challenges and the synchronization techniques that have being proposed for synchronizing video and haptic data.

Comparative tests between H.264 and HEVC are undertaken. Measurements for the network conditions of the Internet are carried out. The equations for the transferring delay of all the inter prediction configurations of the HEVC are defined. Furthermore, it proposes a new efficient algorithm for transferring a real-time High Efficient Video Coding stream with haptic data through the Internet.

Furthermore, it presents the design of a novel real time wireless multisensory smart surveillance system with 3D HEVC features. The proposed high level system architecture of this surveillance system is analyzed. The advantages of the new HEVC encoding are presented. The synchronization issues between the multiple streams are described and solved. All the available wireless standard are presented and compared. A network adaptive transmission protocol for a reliable real-time multisensory surveillance system is proposed. Adaptive Packet Frame Grouping and quantization is enforced in order maximum Quality of Experience to be fulfilled. Measurements from the proposed protocol have given satisfactory results comparing to existing transport protocols.

It also deals with the wireless transfer of real-time high update rate supermedia data over the Internet of Things. It presents the related work on supermedia data transferring and QoE requirements. It proposes a high level architectural design for the transport of wireless multiple supermedia streams over IoT. The most known compression techniques and flow controls for wireless sensory data transferring are analyzed. Based on these compression techniques a new network adaptive flow control algorithm is proposed. Measurements for multihop wireless transferring of high update rate supermedia packets over IoT are presented.

Keywords: Haptics, interactive applications, transport protocols, supermedia, sensors, convergence, converging networks, wireless transmission, HEVC, synchronization

Acknowledgements

I would like to express my deep gratitude to my advisors Prof. Manos Roumeliotis and As. Prof. Kostas Psannis for their patient guidance and enthusiastic support. It has been a great honor working with them.

Last, but surely not least, I would like to deeply thank my wife Katerina, my two daughters Stellina and Aliki, and my family and for their invaluable support.

Contents

List of Figures

List of Tables

Abbreviations

3GSS	Third Generation Surveillance Systems
5G	5th generation mobile networks
AAC	Advanced Audio Coding
ADPCM	Adaptive Differential Pulse-Code Modulation
ADVISOR	Annotated Digital Video for Intelligent Surveillance and Optimized Retrieval
ALPHAN	Application Layer Protocol for Haptic Networking
AMRTP	Adaptive Multisensory Real-time Transmission Protocol
AP	Access Point
BTP	Bidirectional Transport Protocol
CNAK	Cumulative Negative AcKnowledgement
CRA	Clean Random Access
CTU	Coding Tree Units
CU	Coding Unit
CWND	Congestion WiNDow
DPCM	Differential Pulse-Code Modulation
ETP	Efficient Transport Protocol
FPS	frame rate per second
GOP	Group of Pictures
GPB	Generalized P and B Pictures
H5H	HTML5 Haptics
HCS	Haptic Command Station
HECS	Haptic Equipment Control Station
HEVC	High Efficiency Video Coding
HIP	Haptic Interaction Pointer
HMTP	Hybrid Multicast Transport Protocol
HS	Haptic System
IDR	Instantaneous Decoder Refresh
IoT	Internet of Things

IPG	Inter Packet Gap
IRTP	Interactive Real-time Protocol
JCT-VC	Joint Collaborative Team on Video Coding
JND	Just Noticeable Differences
M2M	Machine to Machine
MCS	Motion-Copying System
MOS	Mean Opinion Score
MTU	Maximum Transmission Unit
MU	Media Units
NACK	Negative ACKnowledgement
NAFCAH	Network Adaptive Flow Control Algorithm for Haptic data
NS2	Network Simulator 2 ()
PAD	Packet Arrival Deviation
PCM	Pulse Code Modulation
PDV	Packet Delay Variation
PEC	Prediction and Estimation Component
PEU	Prediction and Estimation Unit
PSNR	Peak-to-Noise Ratio
PU	Prediction Unit
QoE	Quality of Experience
QoS	Quality of Service
QP	Quantization Parameter
RFID	Radio-Frequency IDentification
RMTP	Reliable Multicast Transport Protocol
RTCP	Real-time Transport Control Protocol
RTNP	Real Time Network Protocol
RTP	Real-Time Protocol
RTP/I	Real Time Application Level Protocol For Distributed Interactive Media
RTT	Round Trip Time
SACK	Selective Acknowledgment
SCTP	Synchronous Collaboration Transport Protocol
SNR	Signal-to-Noise Ratio

SRM	Scalable Reliable Multicast
SRTP	Selective Reliable Transmission Protocol
s-SCTP	Smoothed - Synchronous Collaboration Transport Protocol
STRON	Supermedia Transport for Teleoperations over Overlay Networks
SUR	Sensor Update Rate
TFRC	TCP Friendly Rate Control
TU	Transform Unit
UDP	User Datagram Protocol
UHDTV	Ultra High Definition Television
UR	Update Rate
VSAM	Visual Surveillance and Monitoring
VTR	Virtual-Time Rendering

1 Introduction

1.1 Problem Definition

It is widely accepted that the growth of Internet and the improvement of its network conditions helped real-time applications to flourish. Apart from video and sound, new kind of real-time data are making their appearance, such as haptic and other sensory data, all of them known as supermedia data. Haptics refer to the science of manual sensing and manipulation of surrounding objects and environments through the sense of touch [1]. With the optimization of telerobotics and the improvement of Internet status, there is a constant effort to transfer the sense of touch through the Internet. The unstable network conditions, as the end to end delay variation of the Internet, prevent the successful transfer of high frequency haptic data and minimize the Quality of Experience (QoE) of the user. For the successful transfer of supermedia data, specific transport protocols with flow control algorithms should be enforced.

1.2 Motivation and Scope

Motivation. As we are heading towards the 5th generation of wireless/ mobile broadband networks, numerous devices and networks are interconnected. The Internet of Things (IOT) is becoming a reality. Person-to-person, person-to-object, and object-to-object are continuously exchanging massive real-time supermedia data. The efficient transmission of this data is of great importance for the Quality of Service (QoS) that the Internet offers.

Scope. The scope of this thesis is to propose a transport protocol for efficient delivery of supermedia data. Flow/congestion control algorithms and synchronization techniques should be enforced in order to maximize the QoE of a supermedia user.

1.3 Thesis's Contribution

This Thesis presents a survey of transport protocols for supermedia applications. It outlines the related work on these protocols. Moreover, it describes the Quality of Service (QoS) requirements for supermedia applications that a network has to fulfill. It performs a classification of related protocol capabilities and outlines the flow requirements that should be met by protocols designed to carry supermedia data.

Synchronization techniques are analyzed and enforced between multiple supermedia streams. A proposed Haptic system architecture is presented. The efficient synchronization of video, audio, and haptic data is analyzed and performed. The related work on High-Efficiency Video Coding is outlined. A novel algorithm for the efficient transmission of supermedia streams such as HEVC video, haptics and other sensory data is described. A novel network adaptive transmission protocol for supermedia transferring is proposed. A novel smart surveillance system that monitors HEVC streams, depth video, haptics and multisensory data is presented. All the papers published based on the above research are outlined in the appendix A.

1.4 Thesis outline

This chapter has discussed the motivation, the scope, and the main objectives of our work and outlined the major contributions of this thesis. The remaining chapters are structured as follow.

Chapter 2 presents related work on haptic data transferring. Chapter 3 analyzes the high level architecture of haptic data transferring. Chapter 4 discusses the haptic data transmission characteristics. Chapter 5 evaluates the most familiar transport and application layer protocols for haptic applications based on their qualitative features. Chapter 6 performs an evaluation of transport protocols for real-time supermedia data with quantity metrics with simulation techniques. Chapter 7 proposes a network adaptive flow control algorithm for haptic data transmission over the Internet (NAFCAH). Chapter 8 proposes an efficient algorithm for transferring a real-time HEVC stream with haptic data through the Internet. Chapter 9 describes a novel real-time wireless multisensory smart surveillance system with 3D - HEVC streams for IOT. Chapter 10 analyses the transferring of wireless high update rate of supermedia streams over IoT. Chapter 11 presents experiments for real time haptic data transferring over multihop wireless infrastructures. Chapter 12 summarizes the contributions of this thesis and outlines future work.

2 Related Work on Haptic Data Transferring

The material in this chapter was presented in [2], [3], and [4].

2.1 Introduction

In the last decade much interest has been given to Haptic applications. This interest is due to the high Quality of Experience (QoE), that a user perceives when he/she uses a Haptic service [5]. Until recently, Haptic applications were limited in number and found only in High-Speed networks due to high processing and transmission requirements of Haptic data. Personal computer processing capabilities and LAN network speed improvements, in conjunction with the growth of WWW and development of Internet applications, made it feasible to develop applications using a special Internet flow-branch called Haptic Internet and operate applications known as tele-Haptics.

Several interesting studies [6] have shown that the transfer of real-time supermedia applications through the Internet is now possible. Several obstacles, such as the network delay and jitter, may still impede the flourishing of supermedia applications [7]. Apart from delay, the scaling factor in macro-micro teleoperations [8], [9] and the difference in inertia between the master and the slave system [10] can also deteriorate systems transparency [11]. Time-Delay compensation techniques [12], [13], [14] can overcome these barriers while Fuzzy Controller techniques [15], [16] can protect the haptic systems from failure when data transmission is insufficient. The high-computational-cost in Tele-Haptic applications can be moderated with a high-performance computing environment such as a computational grid [17].

Focusing on transmission protocol requirements, studies [18], [19] have shown that Haptic applications are sensitive to network conditions such as network delay, jitter, packet loss, out-of-order packet delivery and increased network congestion. Many methods have been investigated to optimize tele-Haptics transmission. Some of these techniques are multiplexing [20], perception-based data reduction [21], prediction-based data reduction [22], network prediction or network resource allocation [23], Haptic visual aid decorators [24], wave variables [25] and application-centric data transmission protocols.

2.2 Internet-Based Supermedia Applications

The expansion of Internet has led to the emergence of Internet supermedia applications. Tele-Haptics can be used in many areas of our life, like education [26], [27], video games [28], [29], military operations [30], tele-surgery [31], [32] and video enhancement [33], [34]. Moreover, Haptics can aid people with disabilities [35], transform virtual reality to augmented reality [36] and enhance communication between people [37]. Furthermore, a motion-copying system (MCS) [38] can be useful for the digital preservation of motions by skilled experts as a haptic database. Supermedia can also enhance communication between people [37] and upgrade the virtual reality to a promising augmented reality [39].

Since supermedia refers to the many human senses, it follows that supermedia can benefit people with impairments. Haptic devices can help visually impaired people with route navigation and neighboring information [40], [41]. With the help of tactile sensors [42], impaired people can now visit a Haptic-museum from their home and explore all its exhibitions with the sense of touch. Haptic devices can help people with kinesthetic disabilities [43], but they can also improve the movement of humanoid robots like the haptic sensing foot system [44].

2.3 Related Work on Haptic Transport Protocols

Several protocols have been proposed to transfer haptic data. Some of them were created purely for this purpose while others were designed for other interactive applications such as robot tele-operation [45]. The protocols that specialize in Haptic data transmission are SCTP [46], SMOOTHED-SCTP [47], IRTP [48], RTNP [49], ETP [50], and RTP/I [51], ALPHAN [52], HMTP [53]. Some of these protocols, such as IRTP and RTNP were not created specifically for Haptic applications but for other Robotic tele-operations and vision, carried out through IP networks. Both Haptic and robotic applications present the same requirements for data transmission, so that these protocols can be applied to both applications.

ALPHAN [52]: It is an Application Layer Protocol for Haptic Networking (ALPHAN). It operates on top of UDP and can easily be customized with the help of an XML-based description file. It supports prioritization with the help of multiple buffers. It uses three types of messages, partially borrowed from MPEG, namely I packets, known as "key updates" that are sent reliably and P and B packets, that are sent unreliably, known as "normal updates". It includes a timestamp and a sequence number to its packets. It also includes a participant and an object ID. All the above properties force the protocol to use 16 bytes of overhead at every packet.

S-SCTP [47]: Smoothed - Synchronous Collaboration Transport Protocol (s-SCTP) is widely used in haptic applications. It is derived from the SCTP [46] but differs in that it includes a buffer at the receiver’s side to reduce the unwanted effect of jitter. It uses two types of packets: the “key updates” which are sent reliably and the “normal updates” that are sent unreliably. In order to avoid the implosion problem it applies negative acknowledgement. It performs congestion control by scaling the transmission rate, depending on the received Ack and Nack messages. It uses interaction streams to carry “differential” packets. Key updates are placed especially in the first and in the last slot of the stream.

HMTP [53]: It is an efficient Hybrid Multicast Transport Protocol (HMTP) for collaborative virtual environments. It is derived from the combination of four protocols: Scalable Reliable Multicast (SRM) [54], Reliable Multicast Transport Protocol (RMTP) [55], Selective Reliable Transmission Protocol (SRTP) [56] and Synchronous Collaboration Transport Protocol (SCTP). It enforces scalability like RMTP and reliability with NACK approach like SRM. Its architecture resembles a multicast tree, like RMTP. It has two types of messages, “key updates” and “normal updates” in an interaction stream, like SCTP. The packets are sent with three modes of transport depending on their type like SRTP. This protocol focuses on collaborative applications with a lot of users in which case multicast is necessary.

IRTP [48]: Another protocol that applies to interactive applications is the Interactive Real-time Protocol (IRTP). It is connection oriented and is located at the transport layer. For its implementation it imitates the transport protocols TCP and UDP, the TCP for the transport of “crucial data” and the UDP for the transport of the “remaining data”. It implements flow and congestion control with the windows size scheme, same as TCP, and error control with the help of a buffer at the sender's side and two buffers at the receiver's side. The main advantage of this protocol is that it uses a very little overhead of only 9 bytes.

RTP/I [51]: The Real Time Application Level Protocol For Distributed Interactive Media (RTP/I) is also a protocol that can be used for haptic applications. It also uses UDP for the “unreliable data” and TCP for the “reliable data”. It contains a participant identifier, a priority field, a sequence number and a timestamp in its packets. It has four types of messages: “event”, “state”, “delta state” and “state query” messages. Its main drawback is that it is not designed purely for haptic application but for general interactive applications. This means that it has a big overhead of 28 bytes at every datagram in order to support a wide spectrum of applications. Taking into account that the data rate in haptic applications is near 1 kHz, a 224 kbps bandwidth is required only for the overhead information, which is mostly unnecessary.

ETP [50]: The Efficient Transport Protocol (ETP) is designed especially for haptic applications. Its main target is to optimize the available bandwidth by trying to minimize the Inter Packet Gap (IPG) and the Round Trip Time (RTT). For this goal it has six states: FAST DECREASE IPG, LOOK, INCREASE IPG, SLOW DECREASE IPG, STABILITY IPG and STABILITY MAX. By adjusting the IPG it performs congestion control. It uses the UDP protocol for transporting its data and it operates the control and the feedback channel as two independent data flows. The protocol that has almost the same features as ETP is the Bidirectional Transport Protocol (BTP) [57] which is specialized in tele-operated robots and has been developed from the same authors.

RTNP [49]: Another protocol for haptic data transmission is the Real Time Network Protocol (RTNP). It was designed for teleoperation but its main drawback is that it can only be implemented in Unix environments. Its main feature is that it uses priority for its packet. The scale of the priority is designated inside the packets. The priority changes the queue in which the packets stand at the sender's or in an intermediate buffer. If a packet is marked as a “real-time” packet, it gets higher priority.

STRON [58]: The Supermedia Transport for Teleoperations over Overlay Networks (STRON) is a transport scheme that uses forward error correction encoding to provide a reliable and fast transmission service. The main attribute of STRON is that it takes advantage of multiple disjoint overlay network paths (one of them is the IP path) to transport its packets. With the use of Reed-Solomon codes it provides a rather reliable transport service without using acknowledgments and retransmissions. For congestion control it enforces the TCP Friendly Rate Control (TFRC) [59].

Apart from the above protocols, a comparative study should be conducted with most common transport protocols for real-time applications as UDP, RTP, DCCP and SCTP and the generic transport protocol TCP.

UDP: This protocol is in the transport layer and provides the smallest delay, if no congestion occurs in the network. It does not provide reliability and congestion control and it is very poor to Inter-Protocol and Intra-Protocol Fairness. It does not use buffer and produces large jitter. It creates the smallest overhead, 8 bytes, and it is characterized as a “best effort” protocol.

RTP: The Real-Time Protocol runs over UDP and includes a sequence number and a timestamp at its 12 byte header. It is an unreliable protocol with out-of-order delivery. It supports multicast and in conjunction with the RTCP (Real-time Transport Control Protocol) monitors the quality of service.

DCCP: The Datagram Congestion Control Protocol runs over IP. It includes a sequence number, provides bidirectional unicast connections, it is unreliable and includes two congestion control algorithms, TFRC and TCPLIKE.

SCTP: The Stream Control Transmission Protocol runs over IP. It is reliable with an option for order-of-arrival delivery and includes a congestion control algorithm and supports multi-streaming.

TCP: This is a protocol that creates a virtual connection between sender and receiver. It is a reliable protocol with a congestion control mechanism. It creates bigger delay than UDP and it is TCP-friendly with other connections. It uses a bandwidth optimization control with a window size scheme. It creates very big overhead, 20 bytes, that is a disadvantage at a very high update rate. As TCP is optimized for accurate delivery rather than timely delivery, it is understood that it should not be applied in haptic applications.

2.4 Novel Internet Architectures In Regards To Haptic Applications

The growth of www forced the scientific community to come up with novel Internet architectures. Some of these architectures are the IPv6, the HTML5 and the "Internet of Things" [60]. Haptics can be applied to these architectures and benefit from their use.

2.4.1 Benefits from using IPv6 to Haptic transmission

Enforcement of the new Internet protocol IPv6 can improve the transmission of Haptic Data [61]. All clues indicate that IPv6 and QoS will play a major role in tomorrow's communications. All ISP will have to implement the differentiated QoS in their routers and support IPv6.

- IPv6 header contains an 8-bit Traffic Class field. With the help of this octet, Prioritization in data packets can be enforced. This prioritization can help differentiated Qos to be implemented. Haptic data should have higher priority from other multimedia. This necessity derives from the fact that Haptics are real-time data very sensitive to delay, jitter and update rate.
- A field of the IPv6 header that also enhances the real time flow and the management of the QoS [62] is the Flow Label. It is a 20 bit field that informs the router in which flow the packet belongs to and what QoS has to be enforced to. The concept of flows has already been met in haptic and real time protocols as SCTP, S-SCTP, RTP and is known as stream. The help of the flow-stream reduces processing time at the routers, helps packets to travel on the same path and keeps packets in the correct order. Moreover it helps protocols to enforce differential updates, a method that has been met in protocols as [46], [47].
- Another characteristic of the IPv6 is the explicit support of anycast. This feature has already been met in some extensions of IPv4 but it is explicitly supported in IPv6. With the help of anycast transmission a packet can be sent to a group of nodes of which the nearest one is automatically selected. Anycast minimizes the number of hops and the latency of a packet to reach its destination.

- The Multicast transmission is also supported in IPv6 with better bandwidth efficiency than in IPv4. It ensures that all routers support multicasting and offers much larger multicast address space. Haptics can be applied in scalable architectures with multiple users that collaborate with each other. The need of multicast transmission is undoubted. In a large–scale virtual environment multicast group members can be added or removed dynamically. Specific events as new members log in –log out, object acquisition and spam messages should have the ability to be sent to all the participants simultaneously.
- Security and privacy are factors that are being reinforced in IPv6. Haptics could be applied in critical operations like military use and tele-surgery. It is obvious that a crack in the security of a tele-haptic system can cause problems that threat human life.

2.4.2 The Role of Haptics in the "Internet of Things".

It is well known that Internet use is growing exponentially. Apart from the growth of usage, Internet is changing shape, type and services. A few years ago the meaning of "web 2" came in sight. The concept of "web 2" wants the Internet to be a user-centric platform for information sharing. The social media, blogs, wikis and video sharing are in their pick.

New ideas as the "Internet of Things" and the "web squared" [63] are now gaining attention. ''Internet of Things" semantically means ''a world-wide network of interconnected object uniquely addressable, based on standard communication protocols" [64]. The word "object" may refer to all things of our ordinary life such as vehicles, food packages, paper documents, clothes, furniture, etc. "Uniquely addressable" can be done with the help of the Radio-Frequency IDentification (RFID) and the IPv6, since IPv6 provides 128 bits for IP addressing. In the field of "interconnection" apart from RFID, tags, sensors, actuators, semantic technologies and smart phones can provide interconnection, cooperation and a new source of data for the new Internet which is called "web squared". The "web squared" aims to integrate the "web 2" and the sensing technologies in order to provide a more enriched content to users.

Haptics can play a major role in the future Internet called “web squared”. Haptics are the key point of entry into a full sensory virtual reality, called “augment reality”. Haptics can immerse the user and provide the feeling of “being there”. If most objects of the real world addressed uniquely as the “Internet of Things” mentions and modeled/ described with semantic technologies, then with the help of Haptics we can create an “augmented world” which will resemble our physical world. The gap between the digital and physical world will be bridged.

The benefits from such an implementation would be breathtaking. Education, travelling, communications, logistics, robotics and many other areas will change rapidly and will obtain another meaning in our daily life. The “augmented world” will change our descendants’ lives as the Internet and the television changed ours and our antecessors’ lives.

2.4.3 Enforcing the HTML5 technology to Haptic Applications

A new and promising opportunity for supermedia applications to flourish in the Web is HTML5. The HTML5 formal supports the java script based Web Graphic Library WebGL. The display of 3D graphics using a Web browser is now easier. A HTML5 Haptics (H5H) Plug-in [65] runs on most popular web browsers, uses HAPI as a haptic rendering machine and supports most of the commercially available kinesthetic haptic interfaces. The window for easier creation of Web Haptic applications is now opening.

3 High Level Architecture of Haptic Data Transferring

The material in this chapter was presented in [2], [3], and [4].

There are two main categories of system architecture for Haptic data transferring [66]: The server based and the distributed architectures. The client-server architecture has the advantage of consistency. On the other hand, a peer-to-peer architecture supports parallel computation and scalability, and is less dependent on network conditions. The scene in a server-based architecture is only updated after a round-trip delay to the server while in peer-to-peer architecture the update delay is only one-way. Knowing that haptic applications are vulnerable to network delays, in collaborative applications with many participants, a peer-to-peer architecture is often chosen.

A four channel communication architecture for transmitting position and force of the master and slave robot is often proposed [67], [68], [69]. Time delay should be even smaller in crucial tele-surgery operations where the nonlinear - elastic characteristics of soft tissues of the human body complicate system transparency [70], [71]. The 4CH architecture is difficult to be applied as it is dangerous to attach force sensors at the tip of the slave robot.

3.1 Haptic System Architecture

The proposed high level Haptic System (HS) architectural is depicted in Fig. 3-1.

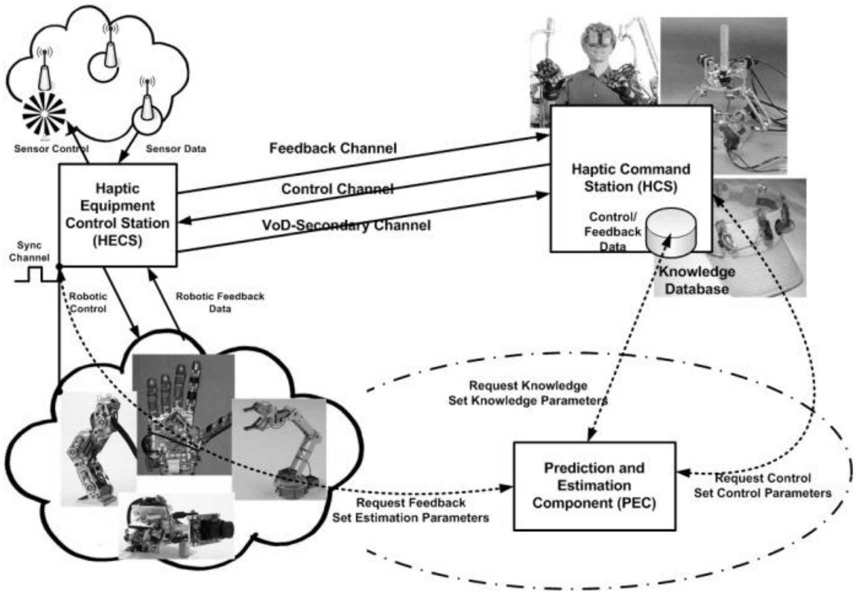

Fig. 3-1: Proposed Haptic system and equipment high level architectural design.

3.1.1 Communication channels

It contains four separate communication channels. These are:

Haptic Control Channel. It carries command queries from the user to the remote Haptic equipment. As Haptic data are very sensitive to network delay and jitter, Haptic data should be transferred by a specific transport protocol for this purpose. Enforcement of strict QoS rules for haptic data should be enforced.

Haptic Feedback Channel. It carries response queries to commands back to Haptic Command Station (HCS) (response info data and sensor data). Response data are both response sensor data of remote haptic devises and supplementary sense information acquired by haptic sensors located at the remote environment. As it transfers Haptic data, strict QoS rules should also be enforced.

HEVC Video Channel. It carries an HEVC encoded video stream from the remote environment back to the user. Depending on the resolution of the video, this stream usually occupies the highest percentage of the bandwidth of the communicational channel.

Audio Channel. It carries audio data from the remote environment back to the user. It is the channel with the smallest QoS requirements.

3.1.2 Structural Units

Synchronization Unit. All the above streams transfer their data through the Internet. Audio, Video and Haptic feedback Media Units (MU) arrive at the Synchronization Unit with disrupted time intervals compared to the generation ones at the source. The main operation of the Synchronization Unit is to preserve the time relation of the original signal as steady as possible and synchronize the three media streams with each other.

Haptic Command Station (HCS). This is the main control station of the HS. It includes a knowledge database for the storage of historical information. Such information could become handy for future use by the Prediction and Estimation Component (PEC) [72], [73]. This PEC component shall either be part of the HCS station or incorporated at a separate management station.

Haptic Equipment Control Station (HECS): It is responsible for managing and monitoring all Haptic hardware equipment and also monitoring and receiving feedback activity from all peripheral sensors that are included in the Haptic system [74].

Prediction and Estimation Component (PEC). PEC is either part of the HCS system or is maintained in a separate system and manages the HCS knowledge database. Its purpose is to monitor the Haptic system commands and provide useful predictions of HCS commands based on feedback data. Such a component shall utilize complicated data mining and estimation algorithms [73]. Its incorporation to the proposed Haptic system and clarification of its requirements and capabilities for certain Haptic system use cases is set as a future work. That signifies the dashed line notation. Placement of the knowledge database either in the HCS or PEC system, focusing on improving Haptic system performance, is also set as a future work.

4 Metrics for the Evaluation of Transport Protocols

The material in this chapter was presented in [2], [3], and [4].

In order to evaluate transport protocols for supermedia applications, analysis, experiments and simulation tests should be carried out. Measures of the performance of a haptic transport protocol should rely both on tangible attributes and on qualitative features.

4.1 Evaluation of Transport Protocols Based on Measurable Metrics

The measurable performance metrics for this evaluation are the packet arrival variation, the delay, the jitter, the packet loss, the packet size and the throughput.

"**Packet Arrival Deviation**" (PAD) is a good metric, proposed by the author, for transport protocol evaluation. In most cases a haptic source tries to send packets at the given rate of 1000 packets per sec. This means that if no congestion occurs and no packets are lost, the packets will arrive at the destination with the same rate. The packet arrival deviation can express not only the delay deviation, known as jitter, but also the packet loss from packets that are dropped during the transmission, the changes of the sending rate of the source and the fluctuations of the Internet bandwidth. The packet arrival deviation cannot only be measured as the standard deviation of the gap of time between the arrivals but also as the max, min and average values of the time gap between the arrivals.

"**Delay**" is the time needed for a packet to travel from the source to the destination. Since the End to End delay can be different for every packet, max, min and average delays have to be measured.

"**Jitter**" is the packet delay deviation which is often expressed as the standard deviation of the end to end delay. This metric is very crucial for the stability of the system. High values of jitter often result in desynchronization, and as a result, the user loses the illusion of "being there" in augmented reality.

“**Packet Loss**” is the percentage of packets that fail to reach their destination. The number of packets that are lost or dropped during transmission increases when congestion occurs. A transport protocol with a congestion control algorithm can significantly minimize the packet loss.

“**Throughput**” is the bandwidth that is consumed from the application in order to send its data to the destination. This bandwidth is derived from the multiplication of the packet size with the packet rate. The optimum packet rate for haptic applications is 1000 packets per sec. Some protocols in order to avoid congestion, lower the packet rate. The packet size also depends on the overhead of the protocol. Given that the payload of haptic applications is relatively small, usually smaller than 64 bytes [48], the lower the overhead is, the more efficient the protocol will be.

“**Efficiency**” [48] is also a metric that has to be measured. The efficiency of each protocol depends on the payload that each application passes to the transport layer and the overhead of the protocol. If the protocol runs on top of the UDP protocol, then the overheads of the two protocols are added. Efficiency is given by Eq. (1). Protocol Efficiency plays a major role especially in applications where the payload is relatively small and the sending rate is high. Such conditions are met in haptic applications.

$$\text{Efficiency} = \text{payload} / (\text{payload} + \text{overhead}). \tag{1}$$

4.2 Evaluation of Transport Protocols Based on Qualitative Features

Using common transport protocol for haptic systems is difficult to successfully transfer the haptic feeling over the network. In order to implement such big update frequencies (1 kHz) with so little delay (50 ms) and jitter (<2 ms), protocols should meet certain constrains and contain some qualitative features. Table 4-1 presents all the protocols that could be used in a haptic transmission system. It also depicts the qualitative features that a protocol should fulfill in order to successfully transfer haptic data through the network. These qualitative features are presented below in a decreasing order of importance:

Reliability: Some data of the control and feedback channel should be sent reliably. The proposed mechanism that reduces feedback amount as much as possible is the Negative Acknowledgement (NACK) [75] and the Selective Acknowledgment (SACK) [76].

Prioritization [77]: Haptic transport protocols should give higher priority to real-time interactive data. If different kinds of data (haptic, video and voice) are transmitted simultaneously, priority should be given to haptic data as they are more sensitive to delay and jitter. As haptic data contain control commands, they are more crucial than video and voice. Video and audio have lower requirements on delay and jitter which means that they should have lower priority. This prioritization can be supported with the IPv4 ToS octet or the IPv6 traffic class octet in a DiffServ architecture.

Key Updates [46]: Many interactive protocols divide packets into key updates which are sent reliably and normal updates which are sent unreliably with best effort techniques. Furthermore, some protocols, like SCTP, support "differential messages". They are update messages that carry only the difference between the current and the previous state of an entity's. This technique achieves higher compression and faster data transmission.

Interaction Stream [46]: In an interaction stream a series of update messages are grouped into one stream. All the messages from the same stream are referred to the same interaction. With this grouping, “differential messages” can be applied, higher compression can be achieved, less bandwidth can be required and less computational power for packet forwarding is needed.

Minimum Overhead: Haptic protocols have to keep as minimum overhead as possible. Haptics require very high update rate, around 1 kHz. This means that each byte at the header is being multiplied by a factor of 1000 due to the update rate. The result of this multiplication is the required bandwidth. Since the packet payload is quite small, usually 32-128 bytes, each byte of overhead sensibly reduces protocol efficiency [48].

Error Correction and Data Integrity: Reliability is a feature that must be supported from a haptic protocol. A small part of the messages, such as the last update message, of the control and feedback channel should be sent reliably. A proposed mechanism that enforces reliability and reduces network traffic is the Negative Acknowledgement (NACK) [75] and the Selective Acknowledgment (SACK) [76]. Both of the above mechanisms avoid the Acknowledgement “Implosion problem”.

Nevertheless both mechanisms increase protocol overhead. A tradeoff threshold should be set in order to satisfy minimum overhead requirement, error correction and data integrity.

Jitter Control [78] and Congestion Control [79], [80]: Jitter control can avoid long delay and jitter due to network overload. When the network is under congestion, data transfer should be reduced or even terminated in order to solve congestion and consequently reduce delay and jitter. Algorithms that could be applied for the congestion control are the Additive Increase/Multiplicative Decrease AIMD [81], the Rate Based Congestion Control RAP [82] and the TCP Friendly Rate Control (TFRC) [59].

Real-time Flow Control: In an IP network, packets can be lost, duplicated, or delivered out of order. A sequence number and a buffer, at the receiver's side will help to set these packets in the correct order. Moreover, the sequence number helps in understanding if any packet has been dropped so that a retransmission can be required. Furthermore, each packet should carry a timestamp. The resolution of the clock must be sufficient for the desired synchronization accuracy. It must allow more precise RTT and jitter estimation. When a packet delays, it means that the network is congested.

Synchronization [83]: As mentioned before, different kinds of data are being transmitted simultaneously over the same channel. Video, voice and haptic media streams have to be synchronized in order for maximum QoE to be achieved. For this synchronization buffer schemes, timestamp fields and time adjustment algorithms are used. It is worth pointing out that media synchronization control is one of the key techniques for realizing distributed multimedia applications. Media synchronization control falls into three types: intra-stream, inter-stream, and group (or inter-destination) synchronization control. Group synchronization control, as well as the first two types of control, is needed in multicast communications. The purpose of the control is to output each MU of media streams simultaneously at different destinations for the fairness among the destinations [84], [85], [86].

Multicast [53]: In the case of a multiuser-collaboration application, the protocol should support Multicast. This property will save users from unnecessary retransmission of identical information from the same source to different recipients. Multicast results in a better utilization of network resources and avoids network congestion. The multicast property can be supported through IP Multicast routing technique.

Throughput adaptation [87]: The protocol should reduce the transfer rate if it detects network overload so as to avoid congestion. In the opposite case, however, if network resources are free for use, it should increase the transfer rate in order to fulfill transmission requirements. Methods for determining an appropriate sending rate of the transport protocol can be found in [88], [89],[90].

Multiplexing [20]: In a haptic application different kinds of data, such as voice, video, graphics and haptic data, are being transmitted simultaneously. Each kind of data has different Quality of Service (QoS) requirements. The statistical multiplexing of this data can achieve better bandwidth utilization.

Scalability- Adaptability [91]: Since network conditions change quite often, the protocol should have the ability to adapt the throughput of the transfer data to the network conditions. When conditions allow it, the protocol should send more detailed information. When a network is heading to congestion, the protocol should have the ability to lower the transferred data either by lowering the data rate or by omitting some packets that are not really necessary.

Receiver Buffer Optimization [83]: Many protocols use buffers to reduce the unwanted effects of jitter and the out of order arrival of packets. With the help of an intermediate buffer the receiver can use packets at the right order regardless of the time of origin. The important drawback introduced by the buffer is that it increases the mean time of the delay. This means that an appropriate receiver buffer length threshold should be applied.

Minimum TCP Friendly Capability: As tele-haptics refer to the Internet, haptic protocols should try to be TCP friendly [92], [93]. There should be a tradeoff between fair network resource allocation and bandwidth optimization so transmission requirements are met.

Table 4-1: Haptic data flow requirements and Transport Protocols Capabilities

Protocol Properties	Protocols									
	A LPHAN	S -SCTP	H MTP	I RTP	S CTP	R TP/I	TP	DP	TNP	CP
LAYER	APP	TRNSP	TRNSP	TRNSP	TRNSP	APP	TRNSP	TRNSP	NTW	TRNSP
PRIORITIZATION	YES	NO	NO	NO	NO	YES	NO	NO	YES	NO
KEY UPDATES	YES	YES	YES	NO	YES	NO	NO	NO		NO
PACKET HEADER (BYTES)	16			9		28		8		20
RELIABLE	PARTIAL	PARTIAL	PARTIAL	PARTIAL	PARTIAL	PARTIAL	NO	NO		YES
CONGESTION CONTROL	NO	ACK-NACK	NACK	Cwnd	ACK-NACK		RATE BASED	NO		Cwnd
BANDWITH OPTIMIZATION	NO	NO	NO	YES	NO	PARTIAL	YES	NO		YES
SEQUENCE NUMBER	YES	YES	YES	YES	YES	YES	NO	NO		YES
TIMESTAMP	YES	YES	NO	NO	NO	YES	NO	NO		YES
BUFFERS	YES	YES	YES	YES	NO	PARTIAL	NO	NO		YES
CONNECTION ORIENTED	NO	YES	NO	YES	NO	PARTIAL	NO	NO		YES
TCP FRIENDLY	NO	NO	NO	YES	NO	PARTIAL	YES	NO		YES
MULTI-PLATFORM	YES	YES	YES	YES	YES	YES	YES	YES	NO	YES

5 QoS Requirements For Haptic Transmission

The material in this chapter was presented in [2], [3], and [4].

The transmission of Haptic data has some unique characteristics that distinguish it from other transmitting multimedia like video and audio. Since Haptics refer to a human sense, it is quite difficult to infer the requirements for transmitting Haptic data. In order to establish the requirements for the Quality of Service (QoS) for supermedia applications, some rather interesting experiments took place [20], [5], [94], [95], [96]. Most of these experiments have as a basic metric the Mean Opinion Score (MOS) [97]. MOS merges the gap between the QoS and the Quality of Experience (QoE). MOS evaluates the quality of experience that a user receives when he/she uses a service. It is observed that the Quality of Experience deteriorates mainly when the network delay is relatively high and unstable. More measurable metrics, as the time that is required to perform specific tasks [98] and the deviation of the Haptic device from a predetermined route [99] have also been used for establishing QoS parameters for Haptic transmission.

The packet delay deviation, often called jitter, is the main reason for the instability of supermedia systems. High values of jitter and delay often occur in best effort, heavily congested networks. Such a network is often the Internet.

Haptic data, as all real-time multimedia streaming applications, require timely delivery of information and can tolerate some packet loss to achieve this goal. When delay and jitter get values higher than a threshold, then malfunctions such as unpleasant sensation of handling, delayed response, oscillations and instability occur. To avoid all these unwanted malfunctions and maximize the QoE of the user, network conditions should satisfy the limitations of Table 5-1 [20], [5], [94], [95], [96]. It is understood that the preferred network conditions vary from application to application.

Table 5-1: QoS Requirements For Supermedia Applications [20], [5], [94], [95], [96].

QOS	***HAPTICS***	***VIDEO***	***AUDIO***	***GRAPHICS***
JITTER (ms)	≤ 2	≤ 30	≤ 30	≤ 30
DELAY (ms)	≤ 50	≤ 400	≤ 150	≤ 100-300
PACKET LOSS (%)	≤ 10	≤ 1	≤ 1	≤ 10
UPDATE RATE (Hz)	≥ 1000	≥ 30	≥ 50	≥ 30
PACKET SIZE (bytes)	64-128	≤ MTU	160-320	192-MTU
THROUGHPUT (kbps)	512-1024	2500-40000	64-128	45-1200

The requirements for the control and the feedback channel are exactly the same. Comparatively speaking, Haptic systems are more sensitive to network delay and jitter, very demanding to high update rate and tolerant to data loss and bandwidth. According to Table 5-1 the refresh rate of the video data is much smaller than that of the Haptic data. The throughput of a Haptic System is smaller than a Video system and a little bit bigger than that of an audio System.

5.1 End – to – End Delay Requirements

The end-to-end delay of a network specifies how long it takes for a packet to travel across the network from the source to the destination. From Table 5-1 it is understood that haptic applications are more sensitive to delay than other common multimedia applications such as Audio, Video and Graphics. The delay in haptic applications has to be smaller than 50 ms while Video, Audio and Graphic applications must have network delay smaller than 400, 150 and 100 ms respectively. The effect of time delay on bilateral teleoperation system stability has been studied on [100]. In order to avoid this difficulty the Impedance-Shaping technique is proposed [101]. One solution that reduces the time delay effect in teleoperation systems is passive bilateral control using wave variables [13]. As for time delay and jitter, we also need some QoS control such as media synchronization control and prediction control [102].

5.2 Jitter Requirements

Haptic applications are prone to instability. This instability is caused by the Packet Delay Variation (PDV), also called jitter [103]. Jitter has to be so small, less than 2 ms, that haptic applications often result in system failure due to instability and lack of synchronization. One compensation technique that partially overcomes this barrier is a buffer that is being put at the receiver's side in order to deliver the packets to the receiver with a more constant rate. The drawback of this technique is that it raises the mean End to End delay.

5.3 Update Rate Requirements

The update rate in Haptic applications should be relatively high, much higher than the refresh rate of the video data. In order to have the maximum sensation of “tele-presence” the update rate should be 1000 packets per second. This high update rate is needed especially in situations where users/objects come in contact with each other. The harder the objects are, the higher the update rate should be, so that unwanted intrusions and oscillations can be avoided [101]. Although interesting studies have been made for the reduction of this high update rate [104], [105], [106], it is still considered to be significantly high. The reduction of the sending rate is mainly based on Weber’s Law of Just Noticeable Differences (JND) which describes the haptic human perception, and is often called the “deadband control” approach [107].

5.4 Throughput Requirements

On the other hand, the throughput in Haptic applications is not very demanding. The small throughput is based on the fact that the packet size of control commands and haptic feedback is relatively small. It is often smaller than 64- 128 bytes [48]. This means that for an update rate of 1000 packets per second the throughput is only 512-1024 kbps.

5.5 Data Loss Requirements

Haptic applications are quite tolerant to packet loss. The data loss in haptic applications may be quite high, up to 10%. The high update rate often compensates the missing packets and the packet loss event is not understood by users. Some transport and the application layer protocol divide packets into normal and crucial updates. Most of the packets are “normal updates” and are sent unreliably. Some packets are characterized as “crucial data” and are sent reliably [52], [47], [48], [51]. The high percentage of the packet loss, 10%, corresponds to the “normal updates”.

6 The Network Conditions of the Internet

The material in this chapter was presented in [3] and [108].

A lot of research [109], [110], [111] has been conducted for the network conditions of the Internet. Network conditions refer to the amount of traffic that is being transferred through the Internet, the End to End delay, the jitter, and the available bandwidth for data transport.

All the above metrics vary in time and space. They depend on the number of the online users, the amount of data that is being transferred at the specific moment of the measurement, and the available equipment of lines and routers. It has been recorded that the amount of data transferred through the web is constantly increasing [112]. Apart from that, the number of online users is increasing as well. The growth of data transfer is compensated by continuing infrastructure upgrades of computer networks.

6.1 Network Monitoring Techniques

There are two types of approaches for monitoring the network status. The two disciplines of network monitoring are the active and the passive measurements [109]. In the active measurement, specific generated probe packets, ICMP messages, are sent to specific destinations; measurements for delay, round trip time, jitter and packet loss are made. Some common diagnostic tools for active measurements are the ping, traceroute, capprobe, pathchar, netem and dummynet [110].

On the other hand, the passive approach monitoring is based on the observation of the traffic that flows on the links. Some passive monitoring tools, commonly called sniffers, are the Tcpdump, Wireshark, Ethereal, Netflow and JFlow [109].

6.2 Internet Network Measurements

In order to infer the network conditions of the Internet, measurements of the average and the standard deviation of the delay, the packet loss rate, and the number of hops of networks between cities, countries, and continents have to be made. 3000 ICMP packets were sent through the Internet between destinations in different regions. Measurements for the above metrics were also made between Japan and Korea, and between Japan and Greece in [113]. A recent measurement that has also been made was between two cities of Greece, Grevena and Thessaloniki. The distance between the two cities is 170 Km. In order to eliminate the relevance of the time of day, 3000 ICMP packets for each of 0, 6, 11, 15 and 19 o' clock standard time were sent from one destination to the other. In order to eliminate the relevance of the client ISP connection of the two destinations, we sent the ICMP packet through two different ISP. In the first experiment, the first client connection was a simple adsl connection of 24 Mbps, while the other client was connected to a private optical network, GRNET [114], part of the pan-European GEANT network with speeds up to 4×10 Gbps. In the second experiment both of the clients were directly connected to the private optical network, GRNET. We managed to measure the mean end to end Delay between the two destinations, the Jitter, the packet loss, and the number of Hops between the two destinations. The results of these experiments are shown in Table 6-1 and Table 6-2.

Table 6-1: Internet Network Conditions for intercontinental communication [113].

COUNTRIES CONNECTED	***AVG. DELAY (ms)***	***Standard DELAY Deviation(ms)***	***PACKET LOSS (%)***	***No. HOPS***
JAPAN - KOREA	27.01	0.19	0.02	11
JAPAN - GREECE	331.10	6.30	1.53	26

Table 6-2: Internet Network Conditions Between Thessaloniki- Grevena, Greece

CONNECTED CITIES	***AVG. DELAY (ms)***	***Standard DELAY Deviation(ms)***	***PACKET LOSS (%)***	***No. HOPS***
GREVENA – THESSALONIKH THROUGH GRNET [114]	19.12	1.70	0	5
GREVENA – THESSALONIKH THROUGH ADSL LINE	53.19	5.31	0.11	8

It is understood that the Internet connection between Japan and Korea satisfy all the restrictions of Table 5-1 for transferring supermedia data through the Internet.

For the Internet connection between Japan and Greece, the values of Table 6-1 are relatively high because they refer to intercontinental pings. The average delay exceeds the limit of Table 5-1. This is due to the fact that the physical distance between Japan and Greece is much larger than Japan and Korea. That's why the number of hops is much bigger in this intercontinental connection.

For the Internet connection between the two cities of Greece the results are shown in Table 6-2. In the case of the simple 24 Mbps Adsl connection the results are slightly worse than the limits in Table 5-1 for the average delay and jitter. The packet loss is within the limits of Table 5-1. On the other hand in the case of the private optical network, GRNET, the results are encouraging. The average delay is only 19.12 ms , the jitter is only 1.70 ms and packet loss is 0.00 %. All the above results are much lower than the limits in Table 5-1, which means that the transport of supermedia data through the Internet is feasible under some circumstances.

Another important factor that describes the network conditions of the Internet is the connection speed of the end user. The recent spread of ADSL and VDSL connections provide consumers with connections up to 50 Mbps bandwidth. This bandwidth is very sufficient for the requirement of 1 Mbps throughput that is being produced from haptic applications, based on Table 5-1.

We can conclude that the Internet network conditions are now suitable for supermedia applications, especially when these applications take place in near regions.

7 Performance Evaluation of Transport Protocols for Real-Time Supermedia - HEVC Streams Over the Internet

The material in this chapter was presented in [3].

7.1 Introduction

Real-time supermedia streams try to transfer audio, video, graphics, haptics, smells and other sensory data over the Internet. As Internet is moving towards the Internet of things, supermedia data obtain massive variety and volume. This increase of data deteriorates the network status of the Internet. An encouraging sign for the Internet's future status is the new video encoding standard HEVC. It offers 50% improvement in video compression over the existing H.264 Advanced Video Coding standard, keeping comparable image quality, at the expense of increased computational complexity [115].

As already mentioned, quite a lot of research [20], [5], [94], [95], [96] has been made on the QoS that a network should support, in order to have the maximum Quality of Experience (QoE) [96] in a supermedia application through the Internet. The network conditions in the Internet are not yet stable. They are changing from one area to another and from one hour to another. Internet network conditions mainly depend on the QoS that the Internet Service Provider enforces and the general state of the net. The recent network conditions of the Internet might permit supermedia applications to flourish. This chapter evaluates the most known, existing transport protocols for the transfer of the real time supermedia data, as haptics and HEVC video streams, over the Internet.

7.2 Simulation Scenario

In order to monitor the metrics of chapter 4 and evaluate the transport protocols of chapter 2.3, simulation tests were carried out. The network simulator that has been chosen for these tests is the Network Simulator 2 (NS2) [116]. It is an open source, discrete event simulator with substantial support for protocol evaluation over wired and wireless networks. A lot of common protocols have already been implemented and tested in the NS2. Unfortunately, not many supermedia protocols have been applied in NS2 so far.

One supermedia protocol that has been implemented in NS2 is the ETP. Apart from that, a lot of real-time protocols that could be used for supermedia applications have been applied in the NS2. Some of them are the RTP, the SCTP, the DCCP and the UDP.

All the above protocols were attached to different nodes in the NS2 as shown in Fig. 7-1. The protocol TCP was mainly applied to the simulation scenario as a traffic generator. All the other protocols try to send a stream of packets with a packet rate of 1000 packets per sec. Most of them have a congestion control algorithm and minimize their sending rate in case of congestion. The haptic packet size that every protocol sends is 64 bytes of data payload [48] plus the overhead of the protocol.

The sample video for the HEVC encoding was the mobile_cif YUV series [117] with 352×288 resolution at 24 Hz. The data rate of this video sample after the HEVC encoding with Quantization Parameter QP= 27 and Low-Delay inter-prediction is 642 kbps [118]. This data stream was sent over the RTP protocol with a packet size of 1500 bytes.

The audio stream was sent over the RTP protocol with 128 kbps bit rate, a packet size of 320 bytes and a sending rate of 50 packets per sec.

7.2.1 Static Network Bandwidth, Delay and Internet Traffic

The square nodes in Fig. 7-1 are routers that are connected with each other through the Internet.

The connection speed between these routers is set to 1, 5, 10, 15 and 20 Mbps for each simulation and is stable for the whole simulation period. The Internet bandwidth of 1 Mbps has been chosen so that a fully congested network can be represented. The Internet bandwidth of 20 Mbps corresponds to a network with no congestion. The Internet bandwidth of 5 Mbps corresponds to a network with low congestion. The connection between the server nodes 6 and 7 is regarded to be the Internet bottleneck of the simulation.

The end to end delay in connection between nodes 6 and 7 was set to 5, 10 ,20 , 30, 40, 50 and 60 ms for each simulation and was stable for the whole simulation period.

The 5 ms delay is a very small delay that rarely occurs in Internet connections. On the other hand, 50 ms is regarded to be the upper tolerable limit of delay, based on Table 5-1, that's why the 60 ms is set as the maximum delay of the simulations. Of course, the end to end delay is changing dynamically in the real world connections. The authors deliberately kept the end to end delay constant throughout each simulation so as to study the behavior of each protocol at the specific delays. This would help the researchers decide which protocols are preferable when the characteristics of the network delay are known. Most of the diagrams depicted in this chapter are for an Internet delay of 40 ms which is an acceptable delay, based on Table 5-1, and commonly encountered in the network. The simulations tests undertaken were 5 (scales for the Internet speed) X 7 (scales for the delay) = for a total of 35 tests.

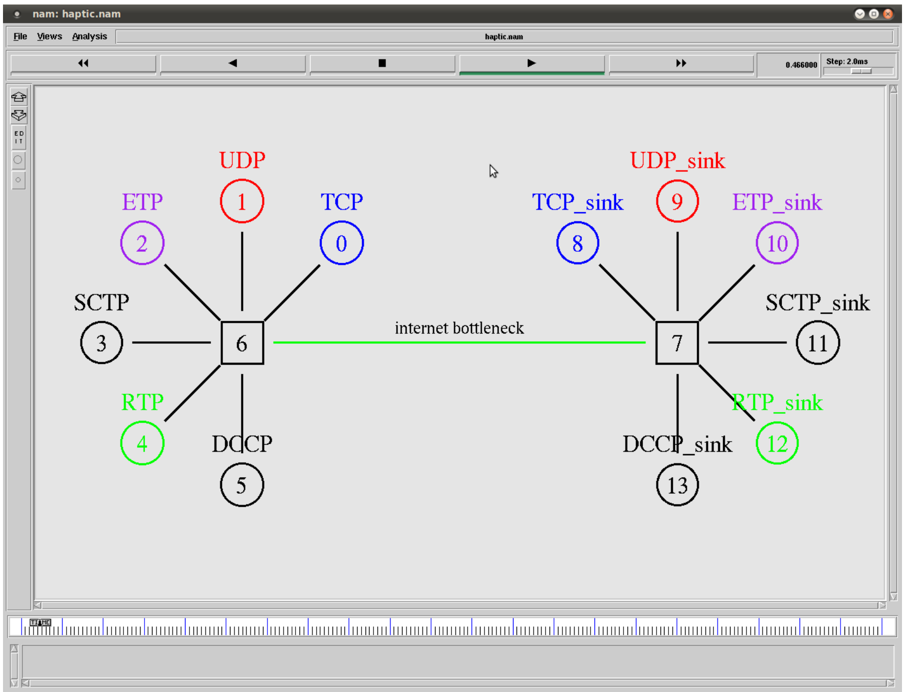

Fig. 7-1: Simulation Environment of NS2.

The connection speed between the nodes 0-5 and server 6 as well as nodes 8-13 and server 7 is 100 Mbps as they are considered to be in the same local area network. The delay in those connections was 1 ms.

The simulation time for each simulation was 20 sec. At time 0.5 sec the FTP application which was attached to node 0 started to send data. At time 2 sec all the other CBR applications which were attached to nodes 1-5 started to send packets with a rate of

1000 packets per sec. The packet size varied from node to node depending on the header of the transport protocol.

7.2.2 Dynamic Network Bandwidth, Delay and Internet Traffic

An interesting case of study is to examine the behavior of all the above protocols in a dynamic environment such as the Internet. In such an environment, the delay of the network, the available bandwidth and the packet loss are constantly changing. In order to simulate such an environment, the DelayBox [119] and the TMIX [120] modules have been added to the ns2 simulator. With the help of these modules, realistic Internet traffic is being fed to the network through the inbound (node 14) and outbound (node 17) initiators of Fig. 7-2. The delaybox nodes on the other hand, enforce a variable bottleneck of 1-20 Mbps, a variable packet delay of 1-20 ms and a packet loss of 0-1% on all the TCP packets that pass through them. As the TCP packets dynamically change their behavior, all the UDP packets are affected as well.

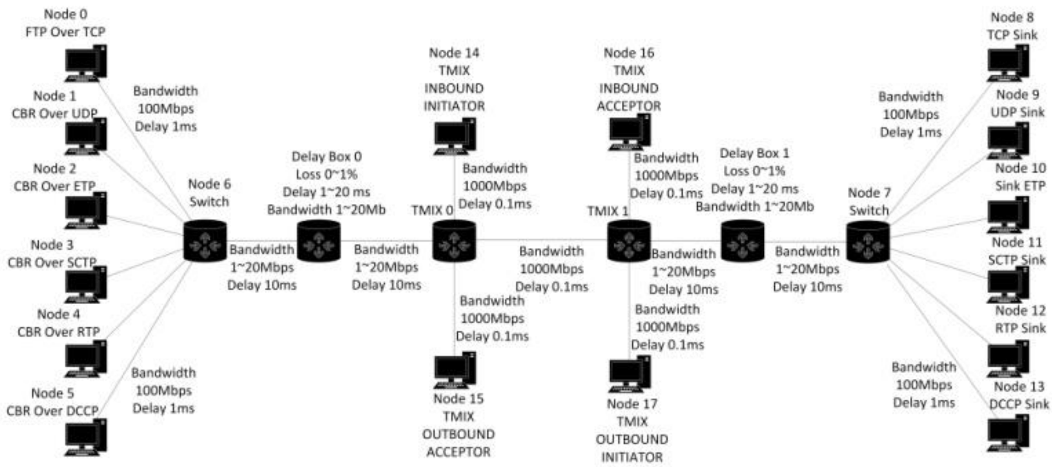

Fig. 7-2: Topology of NS2 with DelayBoxes and TMIX traffic.

7.3 Results And Analysis

7.3.1 Protocol Efficiency

Protocol Efficiency was one of the metrics to be analyzed. As supermedia applications demand very high update rate, the efficiency of the transport protocol is crucial. The small payload should not be overshadowed by big overheads of the transport protocols.

Table 7-1 shows the efficiency of the five protocols that were tested. The protocol with the highest efficiency was, as expected, the UDP protocol. This result derives from the fact that the UDP is a "best effort", unreliable protocol with no congestion control and no packet sorting. Therefore, UDP has the smallest header of only 8 bytes.

Table 7-1: Efficiency Of Transport Protocols

	ETP	***UDP***	***RTP***	***SCTP***	***DCCP***
HEADER (bytes)	12+8(UDP)	8	12+8(UDP)	12+4 (CHUNK INF.)	12
HAPTIC PAYLOAD (bytes)	64	64	64	64	64
EFFICIENCY	76.19%	88.88%	76.19%	80%	84.21%

7.3.2 Packet Loss

Fig. 7-3 and Fig. 7-4 depict the percentage of packet loss in relation to the delay of the network. This diagram is important as it reveals the correlation between the network delay and the packet loss for each protocol. If the characteristics of the End-to-End delay of the network are known, it can be decided which protocol should be used for the transport of supermedia data so as to avoid high values of packet loss.

In Fig. 7-3 the Internet bottleneck is 20 Mbps for all simulations. The delay was set to 5, 10, 20, 30, 40, 50 and 60 ms for each simulation and was stable for the whole simulation period. All the protocols present quite a low packet loss, lower than the limit of 10 % of Table 5-1. The worst performance is presented by the protocols SCTP and DCCP but they still have a packet loss lower than 0.76 %.

On the other hand, Fig. 7-4 presents much higher values of packet loss. In this scenario, the Internet bottleneck has only 1 Mbps bandwidth for all simulations. It is obvious that the network is congested. Six protocols are trying to send a throughput of at least 3 Mbps over the network with bandwidth of 1 Mbps. The protocol with the smallest percentage of packet loss is the UDP protocol. The protocol with the higher packet loss is again the SCTP and the DCCP protocol.

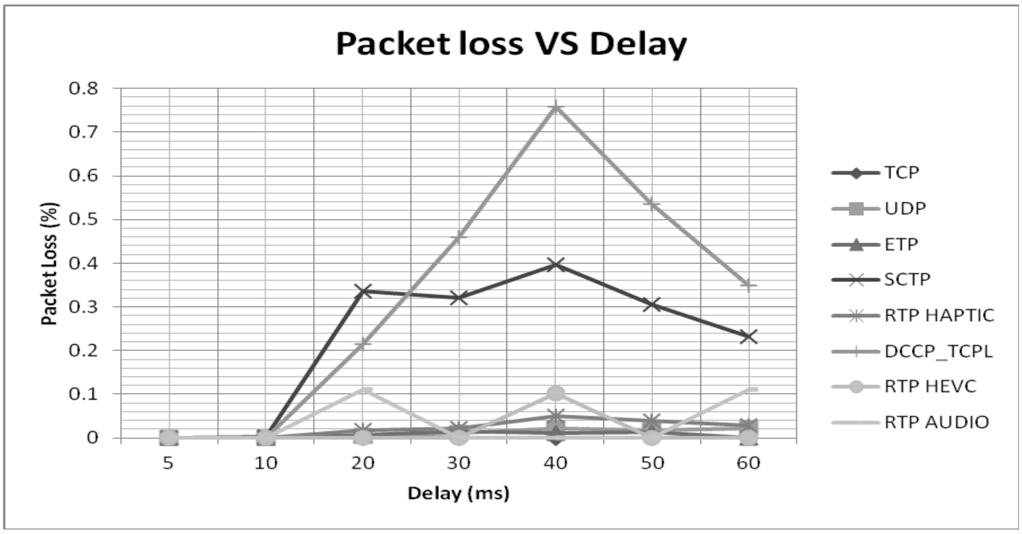

Fig. 7-3: Packet Loss for Internet Bandwidth 20 Mbps.

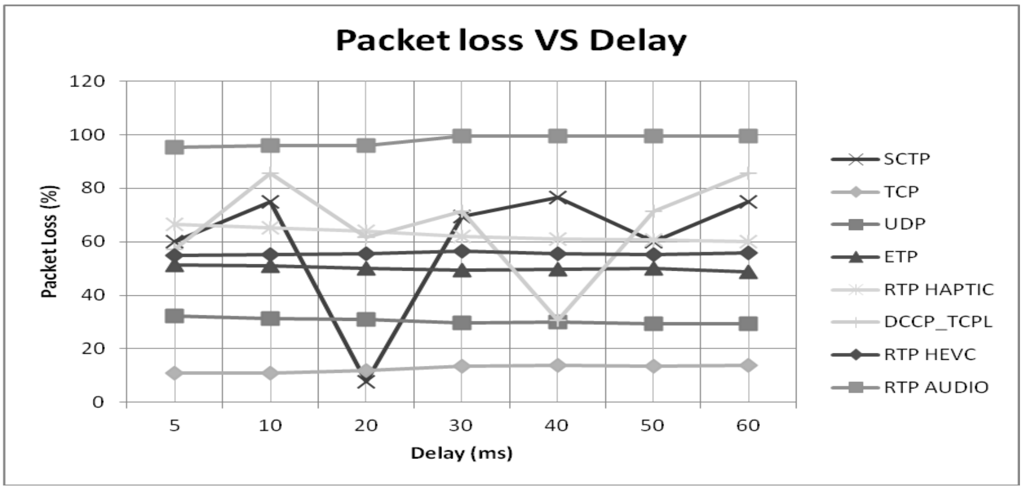

Fig. 7-4: Packet Loss for Internet Bandwidth 1 Mbps.

Fig. 7-5 and Fig. 7-6 depict the correlation between the packet loss and the available bandwidth of the Internet. Fig. 7-5 depicts a network with static network conditions while Fig. 7-6 depicts a network with dynamic delay and Internet bandwidth for the TCP packets. Both charts have similar behavior for Internet bandwidth higher than 5 Mbps, where no significant congestion occurs. From these charts it can be derived which protocols behave better in a congested network. It is quite obvious that as the Internet bandwidth increases, the packet loss decreases. For Internet bandwidth higher than 5 Mbps the packet loss is lower than 1%. This means that there is no congestion on the network for available bandwidth higher than 5 Mbps. When the available Internet bandwidth is only 1 Mbps, the network is heavily congested and the packet losses are unacceptably high. The worst performance regarding packet loss, based on Fig. 7-3, Fig. 7-4, Fig. 7-5 is presented by the DCCP and the SCTP protocol. This performance deteriorates especially in heavy congested networks. The congestion control algorithm of the SCTP and the DCCP protocol is a TCP-like Congestion Control, which is similar to that of TCP. The sender maintains a congestion window and sends packets until that window is full. The response to congestion is to halve the congestion window. This means that the DCCP and the SCTP protocol send their packets in bursts. This causes the buffers of intermediate routers to overflow and some packets to be dropped.

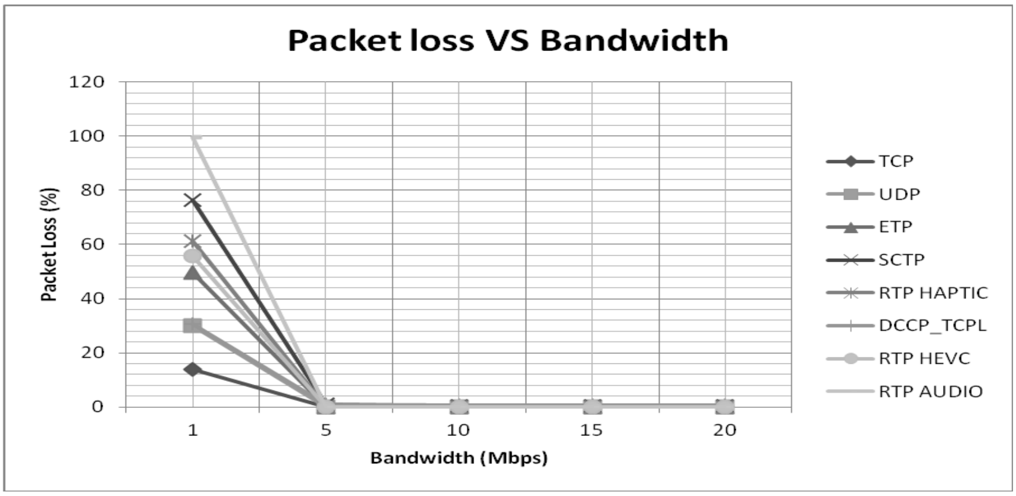

Fig. 7-5: Packet Loss for Internet Delay 40 ms.

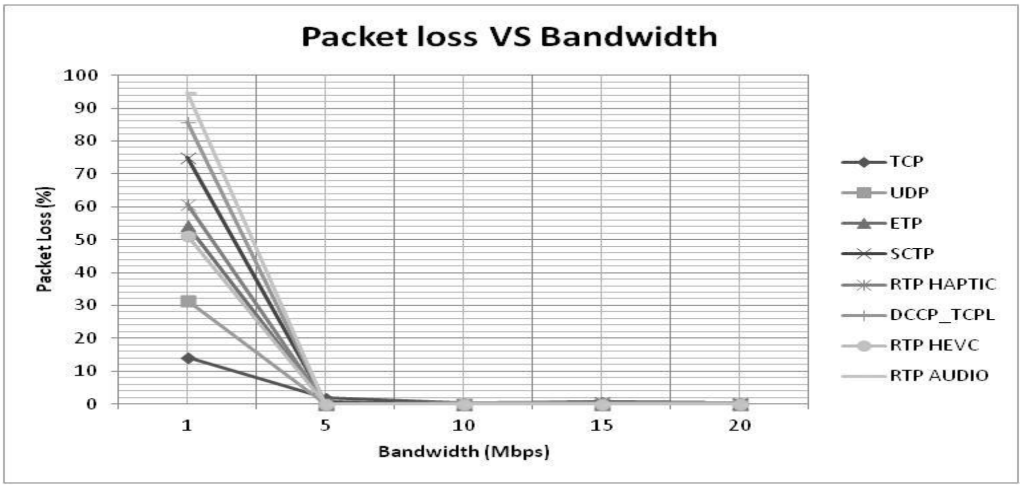

Fig. 7-6: Packet Loss Vs Bandwidth for Dynamic Network Conditions.

7.3.3 Throughput

One more metric that has to be monitored is the throughput that every protocol loads to the network. The higher the throughput is, the higher the possibilities are for network congestion. In connections where the available bandwidth is low, protocols with small throughput should be preferred.

Fig. 7-7 depicts the throughput of the protocols for an Internet bandwidth of 20 Mbps, so that no congestion should occur in the network. The delay of the Internet is set to 40 ms, as 50 ms is the maximum accepted delay, based on Table 5-1. TCP throughput varies between 1660 Kbps and 3300 Kbps. UDP and RTP protocols pose a steady throughput of 576 and 672 Kbps for haptic data respectively. This means that their sending rate is constant. The higher throughput of RTP is due to the higher header of the protocol. Protocol ETP tries to reach its highest sending rate, but its growth is very slow due to its congestion control. After 20 sec of simulation time it had not yet reached the sending rate of 1000 packets per sec. The SCTP protocol presents the highest throughput, after TCP, among the other protocols. Apart from that, it also presents the highest deviation of the throughput. DCCP presents quite a big deviation of the throughput for the first 6 sec and it is not stabilized before the 8th second.

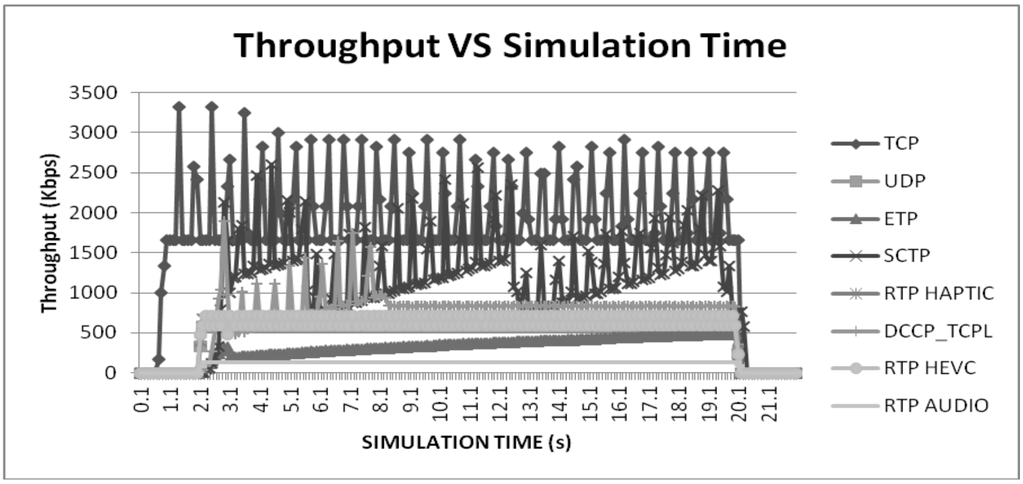

Fig. 7-7: Throughput for Internet Bandwidth 20 Mbps and Delay 40 ms.

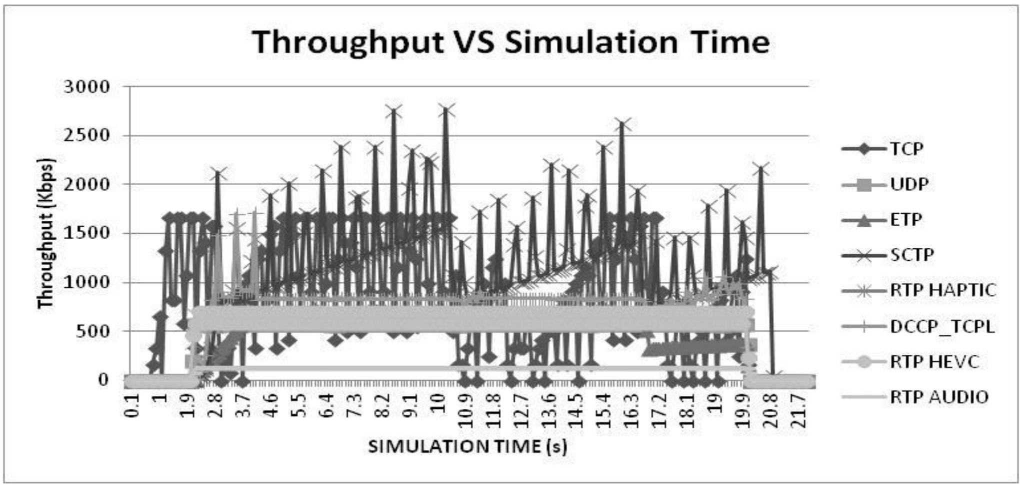

Fig. 7-8: Throughput Vs Bandwidth for Dynamic Network Conditions.

Fig. 7-8 depicts the behavior of the protocols for a dynamic network. The big difference between Fig. 7-7 and Fig. 7-8 is the behavior of the TCP, the SCTP, the ETP and the DCCP protocol. All the above protocols have a tcp friendly congestion control protocol. As the available bandwidth dynamical changes, the above protocols change their transmission rate in order for congestion to be avoided. On the other hand the UDP and the RTP protocol exhibit the same performance because they don’t have a tcp friendly congestion control.

Fig. 7-9 displays the throughput of the protocols for Internet bandwidth 5 Mbps and delay 40 ms. The bandwidth of 5 Mbps is chosen so that the network is under low congestion. TCP has lowered its throughput that now varies from 1660 Kbps to 2160 Kbps. UDP and RTP protocols display almost the same steady throughput with very small deviations as they do not have a TCP-Friendly rate control. ETP protocol adapts the smallest throughput. It cannot increase its sending rate because there is some congestion on the network. It can be seen that the congestion control of TCP binds more bandwidth than that of ETP. DCCP shows almost the same performance with the previous simulation. It adapts almost the same steady throughput, with very small deviations, 2 seconds later than in the previous simulation. SCTP still presents the biggest deviation and does not manage to obtain a steady throughput.

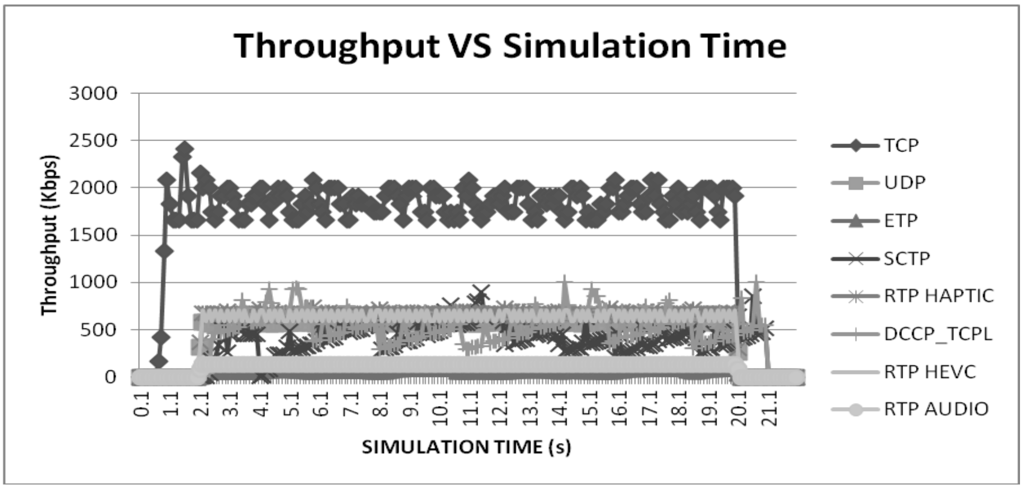

Fig. 7-9: Throughput for Internet Bandwidth 5 Mbps and Delay 40ms.

7.3.4 Jitter

One crucial metric that has to be monitored is the packet delay deviation. A high value of jitter is a crucial factor that often leads to system instability and failure. Based on Table 5-1 haptic applications have the lower tolerant limit of jitter of all multimedia applications. The jitter effect occurs when there is congestion on the network.

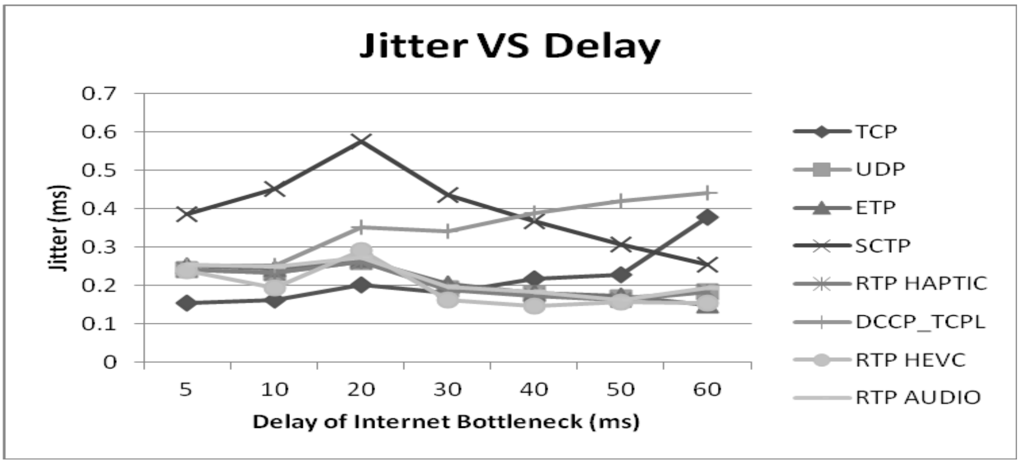

Fig. 7-10: Jitter for Internet Bandwidth 20 Mbps.

Fig. 7-10 illustrates the jitter of the protocols for an Internet bandwidth of 20 Mbps and for delays from 5 ms to 60 ms. At this high bandwidth no congestion occurs. Almost all of the protocols show very small jitter, lower than 0.6 ms.

Fig. 7-11 depicts the jitter of the protocols for an Internet bandwidth of 1 Mbps. It is a fully congested network with a lot of packets lost. The jitter is now much higher than in Fig. 7-10. The protocols with the highest jitter are the DCCP, the TCP and the SCTP. RTP, UDP and ETP protocol have smaller jitter than 6.7 ms, with the ETP exhibiting the best performance with a jitter smaller than 2.6 ms. According to Table 5-1, the jitter should be lower than 2 ms, a goal that most of the protocols could not achieve due to the congestion of the network. An interesting observation is that the protocols TCP, SCTP, and DCCP lower their jitter as the delay of the Internet bandwidth increases. All these protocols have almost the same tcp-like congestion control algorithm.

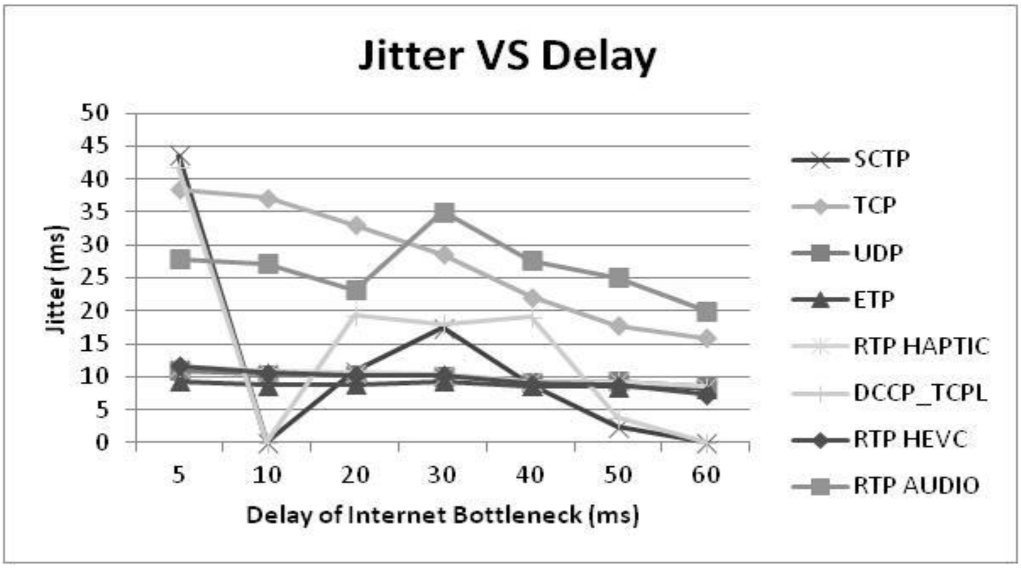

Fig. 7-11: Jitter for Internet Bandwidth 1 Mbps.

Fig. 7-12 and Fig. 7-13 depict the jitter of the protocols when the Internet bottleneck bandwidth varies from 1 Mbps to 20 Mbps. The average Internet delay is near 40 ms. All the protocols show a different behavior when the Internet bandwidth is 1 Mbps and the Network is heavily congested. For higher bandwidth values all protocols present almost the same behavior. As the Internet bandwidth increases, the congestion and the jitter effect decrease.

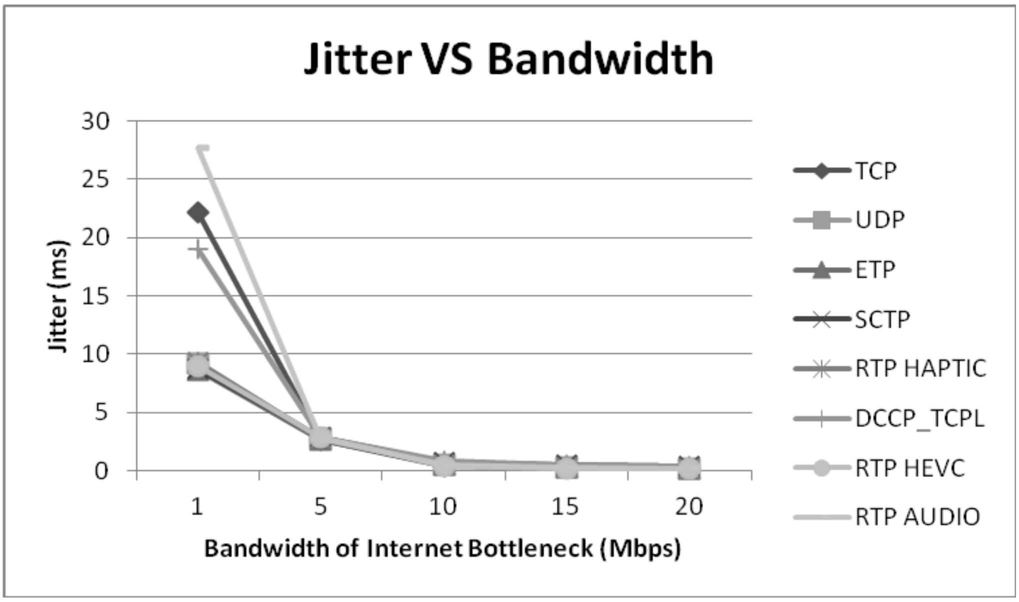

Fig. 7-12: Jitter for Internet Delay 40 ms.

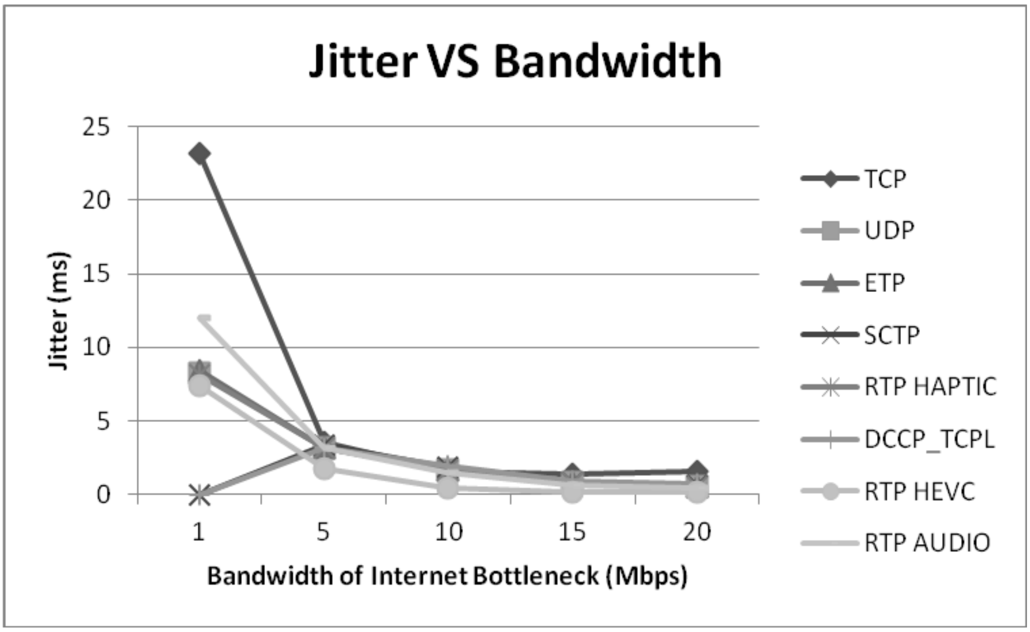

Fig. 7-13: Jitter Vs Bandwidth for Dynamic Network Conditions.

7.3.5 Packet Arrival Deviation

Packet Arrival Deviation (PAD) is a crucial metric for the performance evaluation of supermedia protocols. It shows similar behavior with jitter but it takes also into account the packet loss, the changes of the sending rate of the source and the fluctuations of the Internet bandwidth. Table 5-1 does not include the PAD, as it is only presently proposed by the authors. The upper tolerant limit for PAD should be equal with jitter's, meaning 2 ms, as they should show almost identical behavior when there is no packet loss and no changes in the sending rate and the Internet bandwidth.

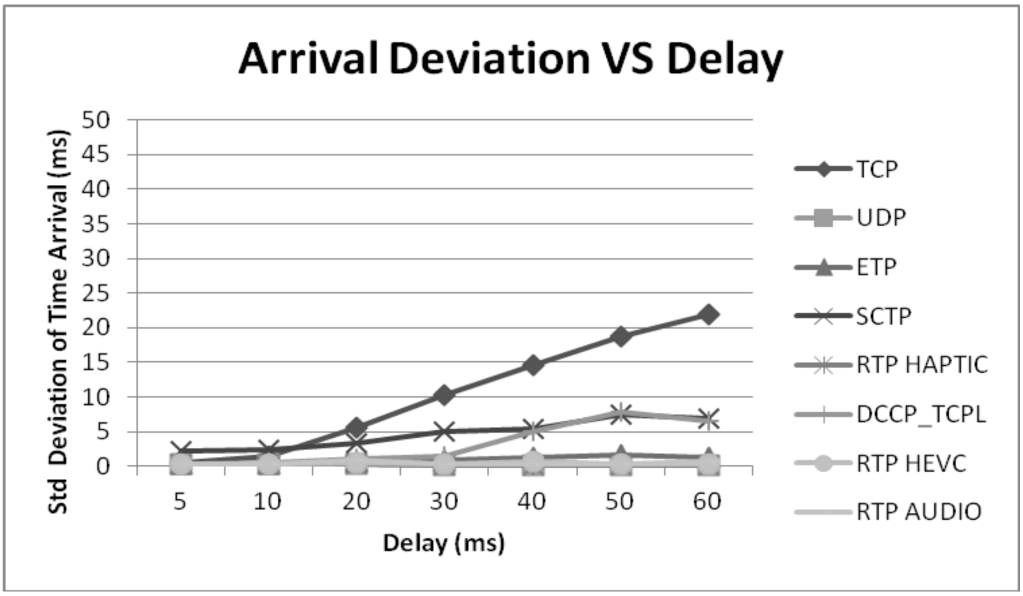

Fig. 7-14: Packet Arrival Deviation for Internet Bandwidth 20 Mbps.

Fig. 7-14 displays the standard deviation of the packet arrival for different delays. The Internet bandwidth is 20 Mbps. The protocols with the higher standard deviation are the TCP, SCTP and the DCCP protocols. UDP and RTP and ETP protocols present very small standard deviation of packet arrival, lower than 2,5 ms. This difference between TCP,SCTP, DCCP and the UDP, ETP,RTP is the way they send their packets. The first group sends its packets in bursts inside a Congestion WiNDow (CWND), while the second group sends its packets with an almost steady inter packet gap.

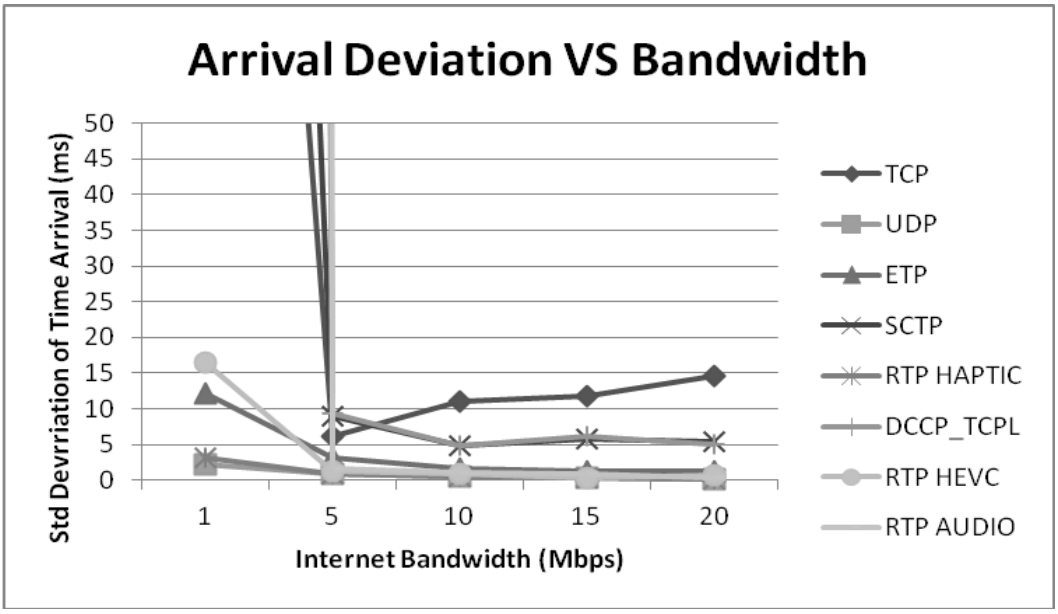

Fig. 7-15: Arrival Deviation for Internet Delay 40 ms.

Fig. 7-15 and Fig. 7-16 shows the Packet Arrival Deviation as the bandwidth of the Internet varies from 1 Mbps to 20 Mbps. Protocols SCTP and DCCP and TCP are not included in the graph for 1 Mbps because they could not perform in such a congested network. The conclusions of this graph are similar to those of Fig. 7-14. The protocol with the highest standard deviation is the TCP, because it sends its packets in bursts. UDP and RTP have almost no arrival deviation as they send their packets with constant bit rate with no congestion control. When the Internet bandwidth gets values higher than 10 Mbps, the performance of all the protocols, except TCP, is similar, as no congestion occurs.

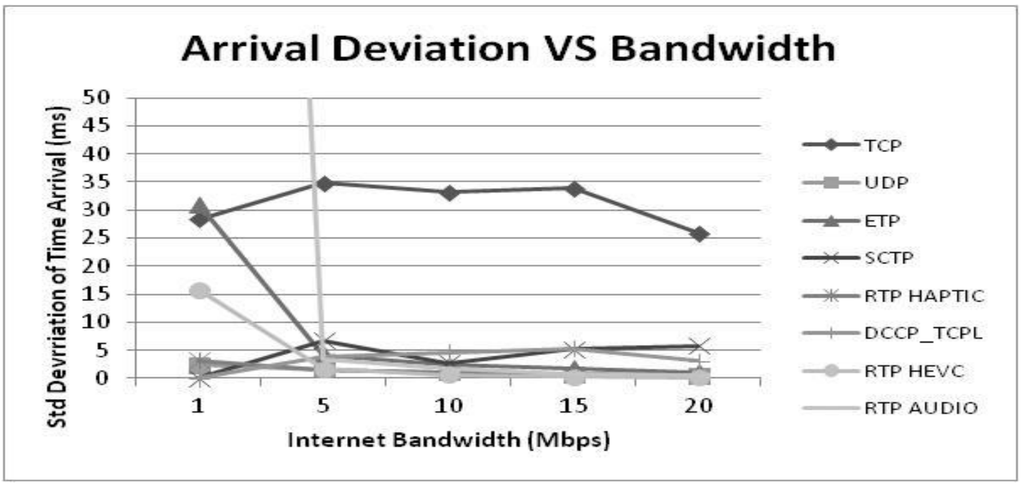

Fig. 7-16: Arrival Deviation Vs Bandwidth for Dynamic Network Conditions.

After analyzing the above results, it is understood that not all multimedia protocols are suitable for transferring supermedia data. Protocols such as TCP, SCTP and DCCP do not perform very well in heavily congested networks, as they are not designed for timely delivery of information. The most stable protocols for real-time data are UDP, ETP and RTP. Both ETP and RTP protocols are based on UDP. The UDP protocol is the lightest, fastest and most efficient protocol from all. Table 7-1 shows that UDP has 88.88% efficiency for a payload of 64 bytes. As far as packet loss is concerned, all protocols show similar behavior, as shown in Fig. 7-5. The only case that they behave differently is when the Internet bandwidth is only 1 Mbps (Fig. 7-4), which means that the network is under unacceptable heavy congestion. As far as throughput is concerned, Fig. 7-7 and Fig. 7-9 which correspond to 20 and 5 Mbps Internet bandwidth with 40 ms delay, UDP and RTP show a steady and similar behavior with UDP consuming a little less bandwidth due to its better efficiency. ETP protocol performs rather well in good network conditions but it can’t reach the optimum sending rate of 1000 packets/sec when a little congestion occurs. Regarding jitter and Arrival Deviation, UDP and RTP show almost the same behavior in all network conditions. Their behavior is better than all the other protocols. The only case where ETP shows a little better behavior than UDP and RTP is when the Internet bandwidth is only 1 Mbps (Fig. 7-11, Fig. 7-12), which means that the network is unacceptably heavily congested. To avoid congestion, ETP is lowering its sending rate, and that’s why it presents lower jitter and arrival deviation. Summarizing, it could be said that RTP and UDP present the best performance with similar behavior in most network conditions. The ETP protocol could be used in cases with network congestion, due to its congestion control.

7.4 Complements, Differences And Relevancies Between Simulation and Real World Experiments

In section VI, our previous real world experiment is described, while in section VII and VIII a simulation experiment has taken place. Anyone would have thought that a real world experiment would give more accurate results than a simulation test and a simulation test is unnecessary. The truth is somewhere in the middle. The simulation tests do not contradict real word experiments but they complement them.

The real world scenario helped us understand the status of the Internet and define the values of the variables for the simulation experiment. The only variable that is difficult to define is the available bandwidth of the Internet for the whole path. It is a metric that is changing dynamically and rapidly as it is based on the number of the online users and the data that is exchanged.

The real world scenario revealed that a real teleoperation task through the Internet is feasible, while the simulation experiment helped us conclude which protocols are better suited for these teleoperation tasks under specific network conditions.

In the real world experiment, it is understood that the experiment results depend on the physical distance between the source and the destination and the ambiguous network conditions of the Internet. In a simulation environment, network conditions are fully controlled, so more accurate results can be produced.

The real world experiment helped the authors define the mean end to end delay, the standard delay deviation (jitter) the packet loss and the number of hops between source and destination. For this experiment the UDP transport protocol was chosen. It is the simplest transport protocol and it is being used for most cases of real-time multimedia applications. Two completely different pairs of source and destination were chosen, in order to examine the dependency between the results on the distance between source and destination.

Despite the ambiguous network condition of the Internet, the simulation results of the UDP protocol match the results of the real world scenario for the connection between Korea and Japan, Table 6-1. At the specific simulation, the Internet end to end delay was set 32 ms. The Internet bandwidth was set to 20 Mbps. These settings depict a network with no congestion. Both simulation and experiment transport data over the UDP protocol. For the connection between Japan and Greece there were no corresponding simulation tests as the results from the real world experiment were outside the acceptable limits of Table 5-1.

Table 7-2: Similarities Between Simulation And Real World Experiments

	DELAY (ms)	***JITTER (ms)***	***PACKET LOSS (%)***
CONNECTION BETWEEN JAPAN - KOREA	27.01	0.19	0.02
SIMULATION OF UDP PROTOCOL WITH 20 MBPS INTERNET BANDWIDTH	32.10	0.19	0.02

8 Network Adaptive Flow Control Algorithm for Haptic Data Over the Internet–NAFCAH

The material in this chapter was presented in [121].

8.1 Introduction

Real time data were considered until recently only video and audio data. With the optimization of telerobotics and the improvement of Internet status, a new kind of data made its appearance in the last decade. This is tele-haptic data. With the word haptics we refer to the tactile and kinesthetic human sense. As human population grows older, the need for teleoperation is getting bigger. With the help of tele-haptics some risky jobs, such as nuclear disposal and wreckage exploration could be done with great safety. Furthermore, applications such as tele-surgery, tele-mentoring, haptic video games, and augmented reality are only few examples of the many sectors of our daily life that tele-haptics could be applied to.

The main obstacle that impedes tele-haptics from flourishing is the delay and the jitter that is being encountered in the Internet. Several congestion/flow control algorithms have been proposed for the limitation of the negative effects of the delay and jitter. Some of them are the TCP congestion window [122], the Additive Increase/Multiplicative Decrease AIMD [123], the Rate Based Congestion Control RAP [82], and the TCP Friendly Rate Control (TFRC) [124], and the variable Inter packet Gap (IPG) [57].

Apart from the common congestion control algorithms some rather interesting techniques intend to reduce the transfer rate of the haptic stream such as the packetization intervals [125], the differential coding with quantization [126], the haptic event prioritization [127], the dead-reckoning [128] and the perception based compression using Kalman filters [129].

Furthermore, the adaptive buffering [128], the haptic packet prioritization [130] and the wave variables [131] try to mitigate the jitter and the delay of the network.

This chapter proposes a network adaptive flow control algorithm for Haptic data over the Internet. The rest of the chapter is organized as follows. Section 2 presents the related work on congestion/flow control algorithms that could be applied to haptic applications and describes network conditions that should be fulfilled for a satisfying QoE. Section 3 analyzes the new proposed flow control algorithm for tele-haptic applications. Section 4 presents graphical representations of experimental data. Finally section 5 identifies conclusions and future work.

8.2 Related Work on Congestion/Flow Control Algorithms

A lot of research has been done for the mitigation of the negative effects of the network delay and jitter. One way to avoid congestion is by minimizing the transmitted haptic packets when delay and jitter show increasing signs [57]. Another method tries to minimize the transmitted packets by forcing the receiver to predict the packets that haven't been received [129]. A further technique transmits only the packets that produce haptic feedback perceptible from human senses [128]. Other methods try to compress the haptic data with lossy data reduction techniques such as the quantization and the differential coding [132]. Furthermore, some researchers send the haptic packets with different priorities [127], the most important packets are sent with higher priority and more reliably than other packets. One more method for minimizing the transfer rate is the packetization intervals [125], where a number of packets are grouped together in a frame and sent to the receiver as a packet.

Each of the above techniques presents some advantages regarding bandwidth, packet loss, and jitter at the expense of precision and average delay. Depending on the application, most of the above techniques improve the Quality of Experience (QoE) of the user in specific network conditions.

8.3 The Proposed Flow Control Algorithm-NAFCAH

Since the network conditions of the Internet are time-varying, the algorithm that controls the transmission of the packets should be network adaptive. Apart from the adaptive transmission rate and bandwidth, priority should be enforced in the haptic packets. Some packets are more important than others. These packets should be sent with higher priority and more reliably. If the network conditions are deteriorating some packets with lower priority should not be sent at all. Another metric that should be network adaptive is the size of the transmitted packets. Techniques such as the differential coding and the quantization, should modify their parameters, in order to change the packet size, and as a consequence the bandwidth of the haptic stream. The transmission rate of the packets should be network adaptive as well. If the network shows some little signs of congestion such as increased delay, jitter and packet loss, then the transmission rate should be reduced in order for heavy congestion to be avoided.

Apart from the maximum acceptable values of the network delay, jitter and packet loss of Table 5-1, intermediate values should be established, in order to escalate the QoE of the users. The maximum values should be escalated into three values. The first scale of these values is [0-max/3] that corresponds to perfect network conditions and the QoE should increase in order to reach its maximum value. The scale (max/3 – 2*max/3] corresponds to fair network conditions where the QoE should increase slowly. The scale (2*max/3 –max] corresponds to acceptable conditions but with high possibility of diversion. The flow algorithm should try to avoid this diversion by keeping the QoE steady. When network conditions are worse than the maximum allowable value of Table 5-1, the flow algorithm should lower the QoE rapidly, so as to avoid congestion. Based on the above assumptions the Network Adaptive Flow Control Algorithm for Haptic data – NAFCAH is proposed.

8.3.1 Network Adaptive Packet Priority

One way to reduce the transmission rate and bandwidth of the haptic stream is to reduce the packets that are being transferred. Packets with lower priority can sometimes be omitted in order to reduce the transfer rate and bandwidth.

8.3.2 Event Priority *ep*.

In [127] the haptic event priority is introduced. When the Haptic Interaction Pointer (HIP) of the haptic interface approaches a virtual object, the haptic packet should obtain higher Event Priority *ep*.

When the distance *d* is between the HIP and the virtual object decreases, the event priority *ep* increases based on the Eq. (2) and depicted in Fig. 8-1. The event priority takes its maximum value *ep=1* when the HIP touches the virtual object.

$$\mathrm{ep} = \begin{cases} 1 & ,\mathrm{dnet} < dmax/3 \\ 1/(\mathrm{n} * \mathrm{dis} + 1) & ,\mathrm{dmax}/3 < dnet < 2dmax/3 \\ 1/(2 * \mathrm{n} * \mathrm{dis} + 1) & ,2\mathrm{dmax}/3 < dnet < dmax \\ 1/(3 * \mathrm{n} * \mathrm{dis} + 1) & ,\mathrm{dmax} < dnet \end{cases} \tag{2}$$

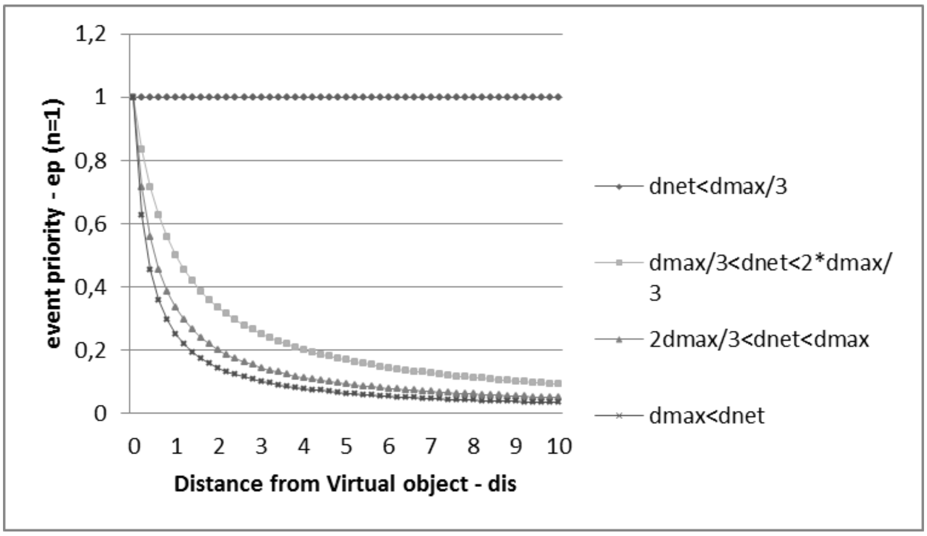

Fig. 8-1: Event priority vs Distance from Virtual object

The higher the priority *ep* is, the higher the importance of the packet to be transmitted. When the priority *ep* is equal to 1, all the corresponding packets should pass the priority check and should be forwarded to the next priority process. The factor *n>0*, is set by the user and represents how steep the curve of Fig. 8-1 and Fig. 8-2 will be. The authors suggest *n=1*. When the factor *ep* is smaller to 1, the HIP doesn't encounter any impedance and the user relies on his/her visual sense to handle the haptic interface. The update rate in this case could be much lower than 1 KHz, equal to the necessary update rate for the vision sense 30 Hz. If network conditions are not adequate, intermediate packets should be dropped. Based on this observation, the update rate of packets that pass the event priority check *pr* is described in Eq. (3) and depicted in Fig. 8-2.

$$\mathrm{pr} = 970 * \mathrm{ep} + 30 \tag{3}$$

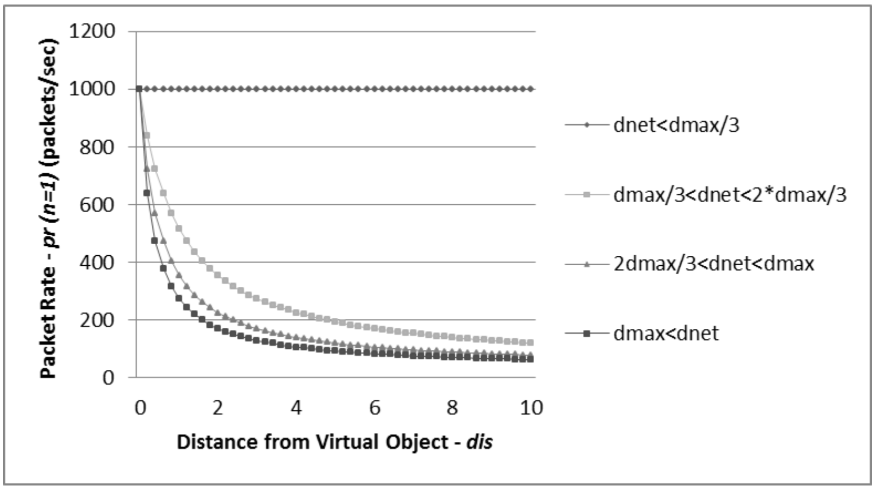

Fig. 8-2: Packet Rate vs Distance from Virtual object

8.3.3 Perception Priority *pep*.

Packets that don't generate perceptible haptic feelings to the user [128] should have lower Perceptible Priority *pep*. The perceptible priority *pep* is based on the dead-reckoning theory [128]. It uses the Weber's law of Just Noticeable Difference (JND) [133] to calculate the threshold ΔI. Eq. (4) calculates the JND based on the stimulus intense I, that the haptic interface causes to the user. The constant κ is called the Weber fraction.

$$\Delta I = I * \kappa \tag{4}$$

When the haptic packet produces a difference on the stimulus intense ***dI*** higher than ΔI then the packet should transmitted with high priority. If the ***dI*** is lower than the ΔI the packet should have lower priority on the transmission.

In order for the algorithm to be network adaptive, the Weber fraction κ_i should change according to the network conditions based on Eq. (5). When the network conditions deteriorate, the constant κ_i should increase in order for fewer, but more important, packets to be transmitted. The factor $0<h<1$ is set by the user and represents how rapid the alteration of κ will be. The authors suggest $h=0,01$. The factor κ_i will try to change its value every time there is a feedback from the network for its network conditions. As far as the Weber fraction κ_0 is concerned Karadogan et al. in [134] have shown that the Weber fraction should take values from $0.08<\kappa_0<0.3$ depending on the application. The smaller the Weber fraction is, the higher the QoE of the user.

$$\kappa_i = \begin{cases} \kappa_0 = 0.08 & \\ \kappa_{i-1} * (1 - 2 * h) & , \mathrm{dnet} < dmax/3 \\ \kappa_{i-1} * (1 - h) & , \mathrm{dmax}/3 < dnet < 2dmax/3 \\ \kappa_{i-1} & , 2\mathrm{dmax}/3 < dnet < dmax \\ \kappa_{i-1} * (1 + 2 * h) < 0.3 & , \mathrm{dmax} < dnet \end{cases} \quad (5)$$

The packets that pass the perception priority filtering pass to the next stage of the sorting.

8.3.4 Prediction Priority *pp*.

Packets that can be predicted by previous packets [126] should obtain a lower Prediction Priority *pp*. Several Interesting studies [129] have shown that most of the transmitted packets could be predicted based on the previous data. If the movement of the HIP is linear the prediction is precise. The prediction unit is installed both at the sender and at the receiver. If the prediction unit at the sender calculates that the current packet could be predicted at the receiver from the last packets that were sent, then the current packet is not transmitted. In order for the algorithm to be network adaptive, apart from the identical predicted packets, predicted packets that are similar to the real packets could be excluded from the transmission. Again this algorithm should be based on Weber's law of the JND to decide which packets could successfully be predicted at the receiver side. If the predicted packet doesn't produce greater difference on the stimulus intense *dI* to the users from the real packet than the threshold $\Delta I'$, then the packet is not transmitted, but it will be predicted on the receiver's side. In this case, the Weber fraction m_i could have different value from the variable κ of Eq. (4). Again the variable m_i should be changed according to the network conditions. The equations that decide which packets are predicted successfully are depicted in (6) and (7). The factor $0<j<1$ is set by the user and represents how rapid the alteration of *m* will be. The authors suggest $j = 0,01$.

$$\Delta Ii' = I * mi \quad (6)$$

$$m_i = \begin{cases} m_0 = 0.08 \\ m_{i-1} * (1 - 2 * j) & , dnet < dmax/3 \\ m_{i-1} * (1 - j) & , dmax/3 < dnet < 2dmax/3 \\ m_{i-1} & , 2dmax/3 < dnet < dmax \\ m_{i-1} * (1 + 2 * j) < 0.3 & , dmax < dnet \end{cases} \tag{7}$$

8.3.5 Adaptive Transmission Rate

Wirz et al. have shown that the network adaptive transmission rate of the haptic stream improves teleoperation [57]. The main obstacle in the variation of the transmission rate is the stable production of packages in the source. If the haptic interface produces update packets steadily and the sender fluctuates the sending rate, a buffer is necessary at the sender side to absorb the fluctuation [128]. The negative aspect of this technique is that if the haptic interface produces packets at a very high update rate (*ur*), usually 1 KHz [135], the sender should transmit its packets sometimes even faster to compensate the previous lower rates. This even higher update rate often results in congestion and packet loss. In order to lower the update rate, an interesting proposal is to integrate a group of packets in a frame and send them as a unified packet, a technique called packetization interval. Fujimoto and Ishibashi [125] have proven that a packetization interval of 8 packets that is sent every 8 ms improves the system's performance in overloaded networks. Another interesting study [130] has shown that the number of the integrated packets np_i should vary, depending on the network delay, so as not to overcome the maximum allowable delay d_{max}. Every packet that is integrated in the frame adds *1/ur* sec of delay. If we take into account the network delay $d_{i,net}$, then the maximum number of integrated packets $np_{i,max}$ is:

$$np_{i,max} = (d_{max} - d_{i,net}) * ur \tag{8}$$

The delay of the network $d_{i,net}$ can easily be extracted if ICMP packets are sent over the UDP protocol from the sender to the receiver. The number of integrated packets is described at Eq. (9).

$$\mathrm{np_i} = \begin{cases} \mathrm{np_0} = 8 \\ \mathrm{np_{i-1}} - 2 > 0 & , dnet < dmax/3 \\ \mathrm{np_{i-1}} - 1 > 0 & , dmax/3 < dnet < 2 * dmax/3 \\ \mathrm{np_{i-1}} & ,\mathrm{2dmax}/3 < dnet < dmax \\ \mathrm{np_{i-1}} + 2 < 8 & , dmax < dnet \end{cases} \tag{9}$$

8.3.6 Network Adaptive Quantization.

The bandwidth that a haptic stream absorbs, depends on two factors, the frame rate, that is determined from Eq. (9) and the size of the frame. The frame size could be reduced if differential coding and quantization [132] is enforced on the packets that are grouped in the frame. The technique that is recommended is the Differential Pulse-Code Modulation (DPCM). In case of a slow motion of the HIP, most of the haptic packets have similar values. The differential coding will produce smaller values than the original ones. Smaller values mean fewer bits. The quantization of the differentiate values could be made with variable quantization step qs_i. The Adaptive Differential Pulse-Code Modulation (ADPCM) for haptic packets was introduced in [132]. Cyrus et al. proposed to alter the quantization step according to the difference size. In this chapter the authors propose to change the quantization step according to the network conditions. When the network conditions deteriorate, the qs_i should increase in order for fewer bits to be required for the reconstruction of the original values. The qs_i is calculated based on Eq. (10). The factor $0<l<1$ indicates how rapid the alteration of qs_i will be, in accordance to the network feedback. The authors suggest $l=0,01$.

$$\mathrm{qs_i} = \begin{cases} \mathrm{qs_0} = 0.3 \\ \mathrm{qs_{i-1}} * (1 - 2 * \mathrm{l}), & < dmax/3 \\ \mathrm{qs_{i-1}} * (1 - \mathrm{l}) & , \mathrm{dmax}/3 < dnet < 2dmax/3 \\ \mathrm{qs_{i-1}} & ,\mathrm{2dmax}/3 < dnet < dmax \\ \mathrm{qs_{i-1}} * (1 + 2 * \mathrm{l}) < 1 & , dmax < dnet \end{cases} \tag{10}$$

The initial quantization step qs_0 in [136] after experiment regarding Mean Opinion Score (MOS) is recommended $qs_0=0.3$ mm.

The flowchart of Network Adaptive Flow Control Protocol -NAFCAH which is based on the above priorities and compressions is depicted in Fig. 8-3.

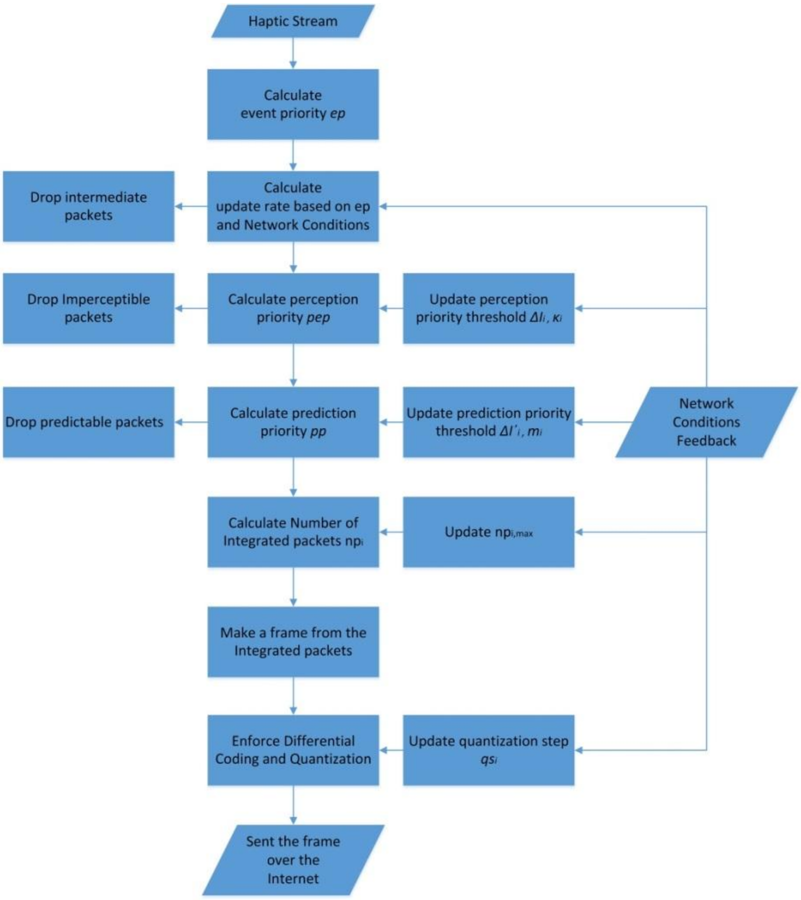

Fig. 8-3: Flowchart of Network Adaptive Flow Control Protocol –NAFCAH

8.4 Experimental Data

The authors, in order to infer the network conditions of the Internet, sent 3000 ICMP packets through the Internet between two cities of Greece, from Grevena to Thessaloniki, 170 km away from each other. Both clients were directly connected to the private optical network, GRNET [114].The results of these experiments are shown in Table 8-1 and in Fig. 8-4.

Table 8-1: A snapshot of the Internet Network Conditions

CONNECTED CITIES	*AVG. DELAY (ms)*	*Standard DELAY Deviation (ms)*	*PACKET LOSS (%)*	*No. HOPS*
GREVENA – THESSALONIKH THROUGH GRNET	31.66	15.7	0.1	6

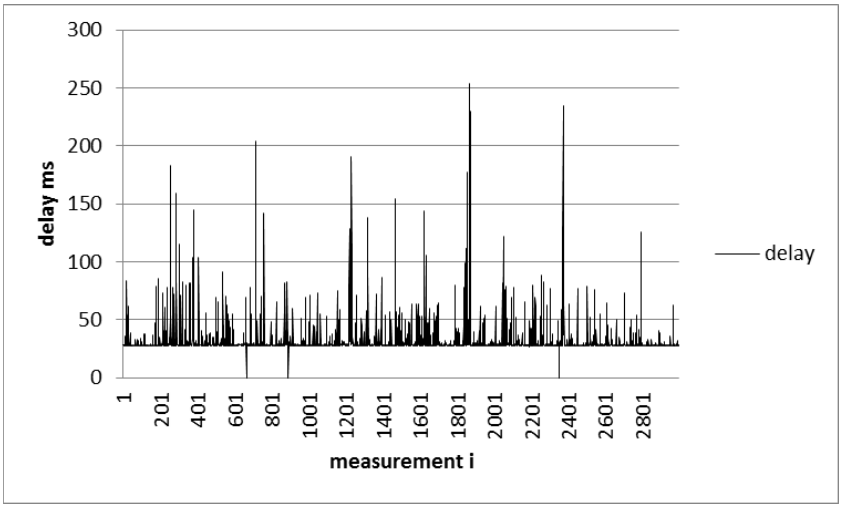

Fig. 8-4: The network delay between Grevena – Thessaloniki, Greece.

The Weber fractions *κ, m* of the Eq. (5) and (7) from the above delay is depicted in Fig. 8-5.

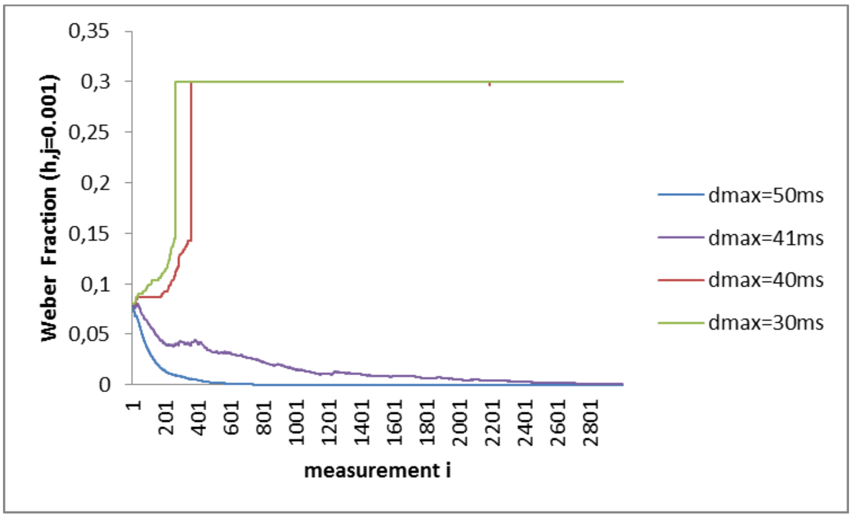

Fig. 8-5: The Weber Fractions κ, m in relation to the network delay.

The Weber fractions start to decrease their value when we set $d_{max}=41$ ms. This means that $2*d_{max}/3= 27.33$ ms. The condition $dnet<2*d_{max}/3$ is being satisfied 972 times, that's why the Weber fractions are decreasing for $d_{max}>41$.

The number of packets *np* of the Eq. (9) in relation to the network delay are depicted in Fig. 8-6.

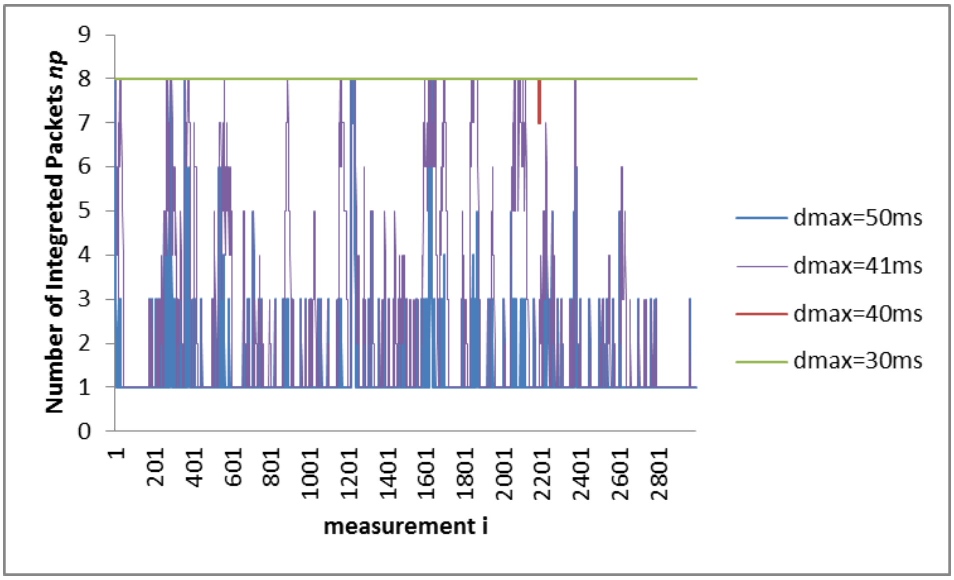

Fig. 8-6: The number of packets np in relation to the network delay.

When the maximum desirable network delay d_{max} is lower or equal to 40 ms, the number of integrated packets is stable at $np=8$. The np is decreasing when $d_{max}>= 41$ ms because the condition $d_{net}<2*d_{max}/3= 27.33$ ms is being satisfied 972 times.

Similarly the quantization step qs is changing according to the network delay based on Eq. (9) and is depicted in Fig. 8-7.

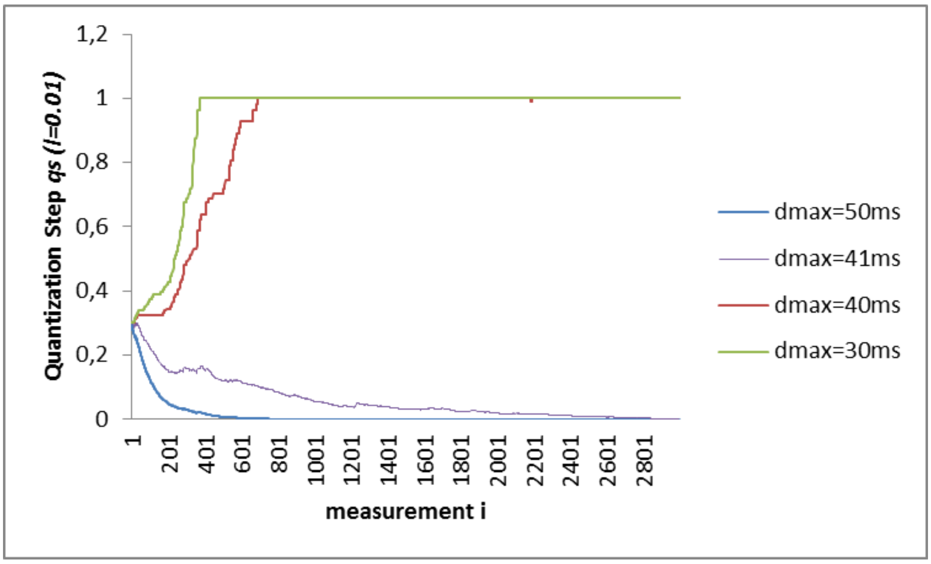

Fig. 8-7: The Quantization step in relation to the network delay.

The quantization step is decreasing when $d_{max}>= 41$ ms because the condition $d_{net}<2*d_{max}/3= 27.33$ ms is being satisfied 972 times. Therefore the quantization step is increasing in order to maximize the QoE. When d_{max}<=40 ms, the condition $d_{net}<2*d_{max}/3$ is never satisfied, therefore the quantization step is constantly increasing.

9 Efficient Algorithm for Transferring a Real-Time HEVC Stream with Haptic Data Through the Internet

The material in this chapter was presented in [118].

9.1 Introduction

The increasing demand for real-time applications with high and ultra-high definition video urged the ITU-T and the ISO/IEC to join their forces to develop the next-generation video coding standard. The Joint Collaborative Team on Video Coding (JCT-VC) has been created. The new coding standard that has been produced is known as High Efficiency Video Coding (HEVC). The proposed HEVC standard fulfilled its target to achieve more than 50% improvement in video compression over the existing H.264 Advanced Video Coding standard, keeping comparable image quality, at the expense of increased computational complexity. HEVC targets a wide variety of high definition video applications such as the 4k television with screen resolution of 4096 × 2160 and the Ultra High Definition Television (UHDTV) with screen resolutions of up to 7680×4320.

The 50% of improvement on video compression that HEVC achieves is denatured in 50% lower bit rate on video streaming. This reduction together with the improvement of Internet network conditions, made the streaming of high and ultra-high definition video over the Internet feasible.

Apart from video, another kind of real-time data which is now trying to travel through the Internet is haptic data. The word haptic derives from the Greek “haptikos” meaning “pertaining to the sense of touch”.

Since haptics refers to the sense of touch, video refers to the sense of vision and audio refers to the sense of hearing, it is becoming clear that all these streams that try to travel through the Internet should be synchronized, in order to achieve maximum Quality of Experience (QoE) for the users.

The rest of the chapter is organized as follows. Section 2 presents the HEVC. Section 3 describes the synchronization algorithms for inter media synchronization. Section 4 analyzes the proposed algorithm for transferring HEVC video stream enhanced with haptic data through the Internet. Finally section 5 identifies conclusions and future work.

9.2 High Efficiency Video Coding (HEVC)

HEVC's main target was to increase data compression by 50% over its predecessor H.264, while keeping the same image quality, at the expense of computational cost.

The image quality can be measured with two kinds of perspectives, objective and subjective. The objective video quality assessment is defined as the signal-to-noise ratio (SNR) and peak-to-noise ratio (PSNR) between the original video signal and the video signal after the encoding and the decoding process. The subjective method is based on the Mean Opinion Score (MOS). Videos are shown to a group of viewers and their opinions are recorded and averaged to evaluate the quality of each video.

Many studies [137], [138], [115], [139], [140] have shown that the increase by 50% of the data compressions has been achieved. Table 9-1 shows the bit rate reduction of HEVC over the AVC for similar video quality, for three videos with different content [138]. Similarly, Jens-Rainer Ohm et al. in [137] showed that the average bit rate saving for entertainment applications is 35,4% measured with the objective PSNR method and 49,3% measured with the subjective MOS method.

Table 9-1: Bit Rate Reduction of HEVC Over AVC For Similar Video Quality [138]

Content	***Objective (PSNR)***	***Subjective (MOS)***
People On the Street	27.5%	50.8%
Traffic	27.5%	74.0%
Sintel2	68.0%	74.7%
Average	44.4%	66.5%

Depending on the application scenario, HEVC offers many configuration modes for efficiency, computational complexity, processing delay, parallelization and error resilience techniques [141].

The two main encoding complexity configurations are the "High Efficiency" and the "Low complexity" modes. The former offers a high efficiency encoding at the expense of computational cost while "Low complexity" offers reasonably high efficiency while trying to keep the encoder complexity as low as possible [142].

As far as the temporal prediction structure is concerned, there are three prediction modes. The first mode is the "intra-only" configuration, where each picture is encoded independently and no temporal prediction is used. The second mode is the "Low-Delay configuration", where only the first picture of the video sequence will be used as an Instantaneous Decoder Refresh (IDR) coded picture, all the other pictures are encoded as Generalized P and B Pictures (GPB), in mandatory Low-Delay test condition, or as P Pictures, which is called non-normative Low-Delay condition. The third mode is the "Random-Access" mode, where the first picture in a Group of Pictures (GOP), which lasts for approximately 1 sec, is encoded as IDR picture and all the other pictures inside the GOP are encoded as B or GPB pictures.

Apart from the temporal prediction, HEVC uses inter and intra spatial prediction based on the coding unit (CU) structure, the prediction unit (PU) and the Transform Unit (TU). Each picture is divided into coding tree units (CTU) of up to 64 X 64 luma samples. CTUs are split into CUs with the help of a generic quad-tree segmentation structure. CUs can be further split into PUs and TUs [143].

Another interesting feature of the HEVC encoder is the slice and tile partition operation. With the help of the slice partitioning, the HEVC manages to fragmentize the encoded pictures near the maximum transmission unit (MTU) size commonly found in IP networks. With the help of the tile partitioning, the HEVC exploits the parallel processing of independent tiles of a picture in multiple cores and processors of a computer [144]. Parallel video coding showed that real-time performance for 1920×1080p/50Hz (53.1 fps) and 2560×1600 (29.5 fps) video resolutions is possible [145].

9.3 Synchronization of Media Streams

One of the negative effects that jitter and delay cause is the desynchronization of the haptic data with the streams of audio and video. This effect is particularly evident in real-time applications that transfer real-time data through the Internet. This is caused mainly by the fact that the end to end delay of each MU is not stable. The MU of each stream may arrive at its destination with a different order than when it left its source [146].

This desynchronization is also accentuated by the following factors:

Each of the voice, video and haptic data stream has different size of MU. The MU of haptic data is usually 40 to 64 bytes. The MU for high efficiency video is usually as the Maximum Transmission Unit (MTU) that the IP protocol can transfer, which is 1500 bytes. The size of the MU for a voice stream is usually 160 to 320 bytes.

MUs are usually transmitted by the User Datagram Protocol (UDP) and the Real-Time Protocol (RTP) [147]. Each of the streams has a different MU rate. The voice data rate is usually 50 MU/sec and the video MU rate is 30 MU/s. Haptic data have a rather big MU rate of 1000 MU/s.

The average bit rate of each stream is different. The video stream, for a 1920×1080 at 24 fps video resolution, has transmission rate, for an HEVC Intra –Low Complexity encoding 4184 Kbps and for an HEVC Low-Delay – Low Complexity encoding 565 Kbps [139]. The average bit rate of the voice stream in case of a linear Pulse Code Modulation (PCM) sound is usually 64 or 128 Kbps. On the other hand, haptic data have, for MU of 40 bytes, an average bit rate of 320 Kbps [146].

In order for all these deviations to be compensated, some rather interesting synchronization algorithms have been proposed [148]. They are divided into two main categories, the Intra-stream and the Inter – stream synchronization algorithms. The former try to preserve the time relation inside a single stream, while the latter try to keep the temporal relation among multiple streams.

Moreover, all the synchronization techniques can be divided into preventive and reactive techniques. The former try to prevent asynchrony, while the latter try to recover asynchrony (skipping, discarding, shortening and extension of output duration, and virtual time-contraction and time expansion) [149].

The evaluation of all the above techniques could be made either subjectively, with the help of the MOS of volunteers, or objectively by measuring the average MU rate, total pause time, average MU delay, and mean square error of inter-stream synchronization [150].

9.4 The Proposed Transferring Algorithm

Quite a lot of research [144], [145] has been done for real-time encoding with an HEVC encoder. All the researchers have come to the conclusion that the real-time encoding with HEVC is feasible, as long as parallel processing with multiple cores is utilized. Apart from the real-time encoding and decoding of the video stream, the audio and the haptic stream should be transferred through the Internet as well.

9.4.1 Synchronization

As mentioned in the previous section, the video, audio and haptic streams have different data rates from each other. This means that the streams are loosely coupled. As a consequence, all the streams have to be synchronized with each other, in order for the maximum QoE to be achieved. Both Intra and Inter-synchronization control should be used.

The intra-stream synchronization keeps the timing relation between MUs of the same stream. It outputs MUs to the destination at the same intervals as the generation ones at the source. On the other hand, inter-stream synchronization tries to reconstruct the temporal relations between the MUs of all the related streams.

The algorithm that is proposed for the synchronization of the three multimedia streams is the enhanced Virtual-Time Rendering (VTR) media synchronization algorithm [151]. The main difference that the enhanced VTR has as opposed to the normal VTR [152] is that the VTR enforces intra stream synchronization on one stream. The enhanced VTR enforces intra stream synchronization on all of the streams separately and an inter synchronization control among the streams. The enhanced VTR has already been enforced between haptic and voice data streams with encouraging results [151].

The first thing that should be defined is which stream is the master stream and which are the slave ones between haptic, audio and video. This mainly depends on the application. If the application is video, audio, or haptic oriented then the master stream should be either the video, the audio, or the haptic stream respectively. If the application has neutral interest among the streams, then the master stream can be derived from the QoS that should be enforced in each stream. From Table 5-1 we can infer that the haptic data are by far more sensitive to delay and jitter than the other streams. This means that the master stream should be the haptic stream. The enhanced VTR will firstly enforce the intra synchronization on all of the streams with the VTR algorithm. Based on the scheduled outputs of the MUs of the haptic stream, it will try to enforce inter synchronization with the other streams.

In the VTR algorithm, the ideal target output time [153] x_n of the n-th MU (n = 1, 2,・・・) is defined as the time at which the MU should be output in the case where jitter is always smaller than an estimate J_{max} of the maximum jitter (that is, the buffering time of the first MU [152]). Let T_n, A_n, and D_n denote the generation time, arrival time, and output time, respectively, of the n-th MU.

The ideal target output time x_n is calculated as follows [153]:

$$x_1 = \begin{cases} D_1(= A_1 + J_{max}), & \text{if } D_1 - T_1 \leq \Delta_{al} \\ T_1 + \Delta_{al}, & \text{otherwise} \end{cases} \tag{11}$$

$$x_n = x_1 + (T_n - T_1) \qquad (n \geq 2), \tag{12}$$

where Δ_{al} denotes the maximum allowable delay.

When jitter is larger than J_{max}, some MUs cannot be output at their ideal target output time. Therefore, the target output time [152] t_n of the n-th MU is introduced, which is calculated by adding/subtracting a delay (i.e., slide time) to/from the ideal target output time. Let t_n^* and ΔS_n denote the modified target output time and the slide time, respectively. Then, t_n and t^*_n are given by

$$t_1 = x_1, \tag{13}$$

$$t_n = x_n + \sum_{i=1}^{n-1} \Delta S_i \qquad (n \geq 2) \tag{14}$$

$$t_n^* = t_n + \Delta S_n \qquad (n \geq 2) \tag{15}$$

where $\Delta S_1 = 0$.

By comparing the arrival time A_n and the target output time t_n, the client determines the scheduled output time [152] d_n ($n \geq 2$) as follows:

$$d_n = \begin{cases} t_n^*, & \text{if } A_n \leq t_n \\ A_n, & \text{otherwise} \end{cases} \tag{16}$$

In the former case of Eq. (16), the target output time is advancing (i.e., the virtual-time contraction, in which $\Delta S_n < 0$); when $\Delta S_n = 0$, we set $D_n = d_n$. The latter case of Eq. (16) delays the target output time (i.e., the virtual-time expansion, in which $\Delta Sn > 0$). In the latter, when multiple MUs have the same scheduled output time, the MU which has the largest sequence number among them is outputted and the other MUs are skipped.

An MU (say the n-th MU) which is not skipped has the output time $D_n = d_n$.

Virtual-Time Expansion

When $d_n - t_n > Th2 > 0$, we set $\Delta S_n = d_n - t_n$, where $Th2$ is a threshold value which we use so as to decide whether the target output time should be delayed or not [152].

Virtual-Time Contraction

When $A_n \leq t_n$, the target output time of the n-th MU is advanced. $d_n = max(t_n - r, x_n, A_n)$ and $\Delta S_n = -min(r, \sum_{i=1}^{n-1} \Delta S_i)$ [153] when $t_n - T_n > \Delta_{al}$, or when a certain period of time (say $T_{nodelay}$) has elapsed since the last late arrival or the last virtual-time contraction, where r is a positive constant. There is a possibility that $d_n \leq D_m$ $(n > m)$ in the case of the virtual-time contraction, where m is the sequence number of the last output MU. In this case, the n-th MU is skipped.

After the calculation of the output time of each haptic MU with the VTR algorithm, the inter-synchronization among the other streams should be enforced. The inter-synchronization will be made at the timestamp that each MU carries. The stream with the highest update rate is the haptic stream, that is 1 KHz, which means that there is an output every 1 ms. Audio and Video streams have much lower output time, which means that the timestamp of their MU could easily be synchronized with timestamps of the haptic MUs.

9.4.2 Temporal Prediction Structure

An important factor that should be taken into consideration for the real-time synchronization of a haptic stream with an HEVC encoded video is the temporal prediction structure that has to be used for the video encoding. As mentioned in section 2, there are three kinds of temporal prediction structures. The first one is the intra-only configuration. Each picture in this kind of configuration is encoded as an Instantaneous Decoder Refresh (IDR) picture. This means that no temporal reference pictures are used. Each frame is independent of the others. A graphical representation of Intra-only configuration is shown in Fig. 9-1. The number next to each frame indicates the encoded and decoded order of the frame. It is understood that in a real-time tele-system, as the frame is captured from the camcorder, it is instantly encoded; it is transferred through the Internet and is decoded from the destination. The negative aspect of this configuration is that it produces extremely high bit rates, which are prohibitive for real-time transferring of data through the Internet.

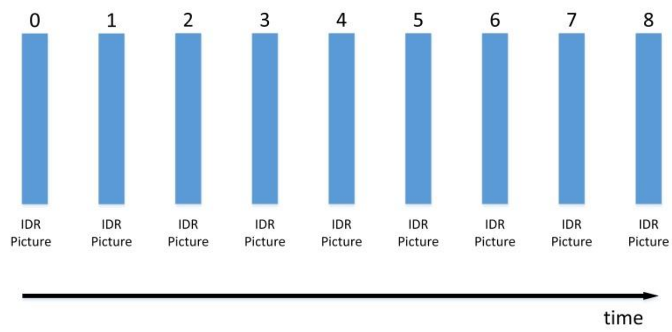

Fig. 9-1: Graphical presentation of Intra-only configuration

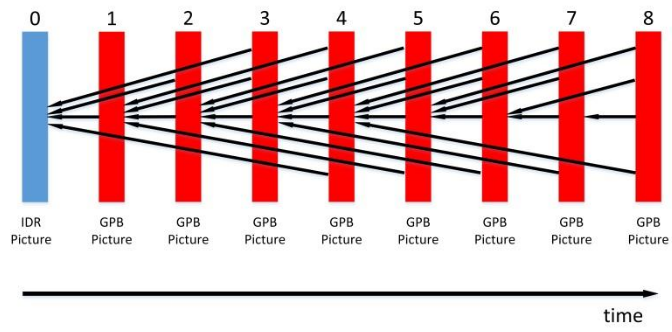

Fig. 9-2: Graphical presentation of Low-Delay configuration

Another temporal prediction structure of HEVC is the Low-Delay configuration. A graphical presentation of the Low-Delay configuration is depicted in Fig. 9-2. The first picture of this configuration is an IDR frame. It is encoded independently. All the other subsequent frames are encoded based on this frame. The encoded and decoded order of the frames is the same as the display order. This configuration is usually proposed for real-time systems as it exhibits the shortest delay of the video transport. The bit rate of this configuration is lower than the intra configuration.

If we assume for simplicity that the delay time of the network between the source and the destination in a remote system is t_{net}, the mean computational time for the encoding of one frame is ten, and the mean computational time for the decoding of one frame is tdec, then the average delay time $t_{Low\text{-}Delay}$ for a video transferring of a Low-Delay configuration is given by the Eq.:

$$t_{Low\text{-}Delay}=t_{en\text{-}LD}+t_{net}+t_{dec\text{-}LD} \qquad (17)$$

The temporal prediction of the Random-Access configuration is depicted in Fig. 9-3. It is understood that the encoded and the decoded time of the frames is different from the display order of the frames.

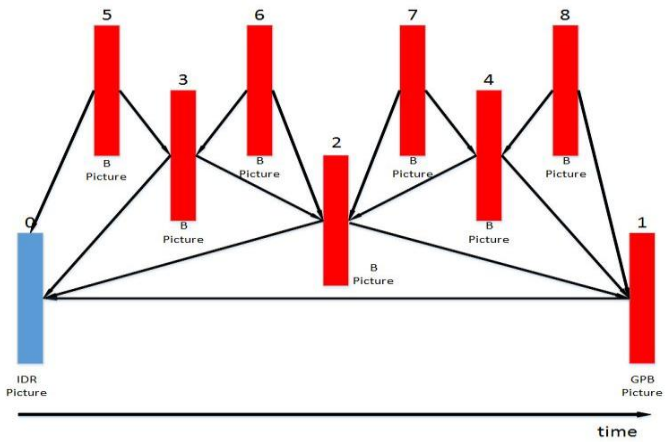

Fig. 9-3: Graphical presentation of Random-Access configuration

The frames are separated in a group of pictures (GOP). If the first frame of the group is encoded as an IDR frame, then the GOP is called closed GOP. If the first frame is encoded as a Clean Random Access (CRA) picture then the GOP is called open GOP.

The strong asset of this configuration is that it demands less encoding time than the other inter-prediction configurations. This feature is very important for applications that transfer data through the Internet. The delay time for this configuration is dependent on the encoded process.

If we assume that the size of the GOP is *gopsize* and the interval time between successive frames is $t_{fr}=1/fps$, the delay time tRandom-Access for a video transferring of a Random-Access configuration is given by the Eq.:

$$t_{Random\text{-}Access}=gopsize*t_{fr}+gopsize*t_{en\text{-}RA}+t_{net}+gopsize*t_{dec\text{-}RA} \quad (18)$$

If we consider the whole encoding, transferring and decoding process as a pipeline, then the delay time $t_{Random\text{-}Access}$ of (18) can be significantly smaller. The first thing that should be determined is the encoding order of each frame. The frame that has to be encoded first is f_0. The second encoded frame is f_{gop}. The third inevitably encoded frame is $f_{gop/2}$. The next encoded frame could be either $f_{gop/4}$ or $f_{gop3/4}$. If we want to save time, then the frame that has to be encoded next is the $f_{gop/4}$. This is due to the fact that the encoded order is the same with the decoded order. If $f_{gop/4}$ is encoded first, it will be decoded first and it will be available for display on the receiver a little earlier. The next frame that has to be encoded is $f_{gop3/4}$. For gop=4 the graphical presentation of Random-Access configuration is shown in Fig. 9-4.

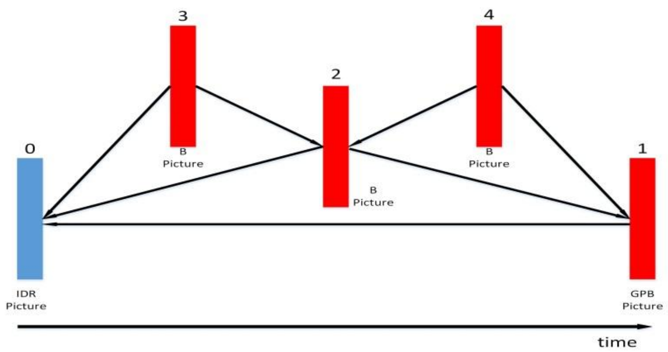

Fig. 9-4: Graphical presentation of Random-Access configuration for gop=4.

The delay time for gop=4 is given by the Eq.:

$$t_{Random\text{-}Access\text{-}4}=3*t_{fr}+3*t_{en\text{-}RA}+t_{net}+t_{dec\text{-}RA} \quad (19)$$

The gop should be a power of 2, so that a hierarchical frame prediction structure could be made. With the same process, for gop=8 the graphical presentation of the GOP is shown in Fig. 9-3. The encoded order of the frames is shown as a number next to the frame.

The system will start to show the first decoded frame f_0, $D1= t_{fr}$ sec before the second frame f_1 of the GOP is decoded. In order for the second frame f_1 to be decoded, all the frames of the GOP have to appear, which means after $D_2=t_{fr}*gop$ time. After all the GOP appear, the encoder will start to encode the frames in the exact order f_8, f_4, f_2, f_6, f_1 as shown in Fig. 9-3 (The f_0 frame will have already been encoded during the D_2 interval). This means that the whole system will wait another $D_3=5*t_{en\text{-}RA}$ secs. When the frame f_1 is encoded, it will be sent through the network to the decoder with network delay time $D_4=t_{net}$. The decoder will be ready to show the f_1 frame after $D_5= t_{dec\text{-}RA}$ decoding time.

The summarized delay time for gop=8 is given by the Eq.:

$$t_{Random\text{-}Access\text{-}8}=-D_1+D_2+D_3+D_4+D_5$$

$$\equiv$$

$$t_{Random\text{-}Access\text{-}8}=7*t_{fr}+5*t_{en\text{-}RA}+t_{net}+t_{dec\text{-}RA} \qquad (20)$$

The delay time of Eq. (20) may be reduced if we change the order of the encoded frames of Fig. 9-3 without changing the correlation between the frames. Since the correlation between the encoded frames is not changing, the encoding time and the data rate of the whole GOP remains the same.

If we try to always encode the left “available” frame of the hierarchical structure, then the encoded order of the frames is shown in Fig. 9-5.

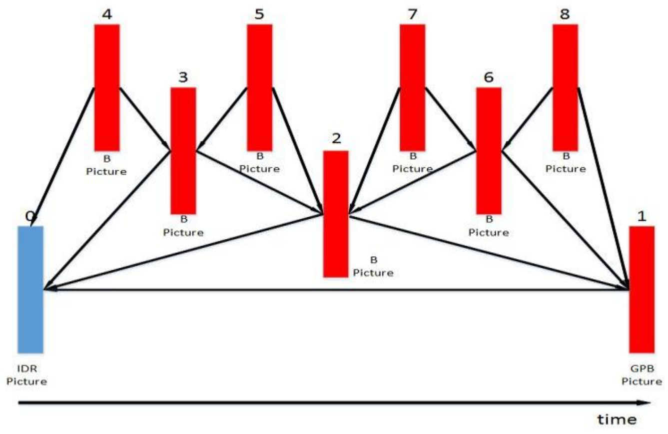

Fig. 9-5: The proposed encoding order inside the GOP=8

The modified delay time for gop=8 is given by the Eq.:

$$t'_{Random\text{-}Access\text{-}8}=7*t_{fr}+4*t_{en\text{-}RA}+t_{net}+t_{dec\text{-}RA} \quad (21)$$

$$dt=t_{Random\text{-}Access\text{-}8}-t'_{Random\text{-}Access\text{-}8}=t_{en\text{-}RA} \quad (22)$$

The delay time is reduced by $dt=t_{en\text{-}RA}$, as given by Eq. (22). This reduction is due to the fact that the encoder will start to encode the frames with the exact order f_8, f_4, f_2, f_1 as shown in Fig. 9-5. This means that $D_3{'}=4*t_{en\text{-}RA}$.

Eq. (21) is generalized for gop as:

$$t_{Random\text{-}Access\text{-}gop}=(gop-1)*t_{fr}+(\log_2(gop)+1)*t_{en\text{-}RA}+t_{net}+t_{dec\text{-}RA} \quad (23)$$

Eq. (23) can be theoretically explained as follows.

When the GOP starts, the first frame f_0 appears. The encoder starts to encode f_0. When the encoding of the frame f_0 finishes, the encoded frame is sent through the Internet to the decoder. Until f_{gop} appears at $t=gop*t_{fr}=gop/fps$, the encoder buffers the frames f_1, f_2,..,f_{gop-1}. When f_{gop} appears, the encoder starts to encode the buffered frames with the following order: f_{gop}, $f_{gop/2}$, $f_{gop/4}$, $f_{gop/8}$, ...,$f_{gop/gop=1}$. In the meantime every frame that is encoded is sent through the Internet to the decoder.

At t_2, the frame f_1 has been encoded, has travelled through the Internet, has reached the decoder and has been decoded.

$$t_2=gop*t_{fr}+(log_2(gop)+1)*t_{en\text{-}RA}+t_{net}+t_{dec\text{-}RA} \quad (24)$$

The frame f_0 should reach the decoder and be decoded at

$$t_1=t_{en\text{-}RA}+t_{net}+t_{dec\text{-}RA} \quad (25)$$

In case of a real-time encoding we have to accept that

$$t_{fr}>t_{en\text{-}RA}+t_{dec\text{-}RA} \quad (26)$$

For the Random-Access configuration we have

$$gop>1 \quad (27)$$

which means that Eq. (26) and (27) lead to

$$\begin{gathered} t_{en\text{-}RA}<gop*t_{fr} \\ \equiv \\ t_{en\text{-}RA}+t_{net}+t_{dec\text{-}RA}<gop*t_{fr}+(log_2(gop)+1)*t_{en\text{-}RA}+ \\ +t_{net}+t_{dec\text{-}RA} \\ \equiv \\ t_1<t_2 \end{gathered} \quad (28)$$

The frame f_0 has to come out of the decoder buffer at least t_{fr} time before the decoding of the f_1 at t_2. Therefore, we should have

$$\begin{gathered} t_1<t_2\text{-}t_{fr} \\ \equiv \\ t_{en\text{-}RA}+t_{net}+t_{dec\text{-}RA}+t_{fr}<gop*t_{fr}+(log_2(gop)+1)*t_{en\text{-}RA}+ \\ +t_{net}+t_{dec\text{-}RA} \\ \equiv \\ t_{fr}<gop*t_{fr}+(log_2(gop))*t_{en\text{-}RA} \end{gathered}$$

$$\equiv$$

$$0<(gop-1)*t_{fr}+(log_2(gop))*t_{en\text{-}RA} \tag{29}$$

Inequality (29) is true as both of the totalizers are positive.

This means that the video stream will start at

$$t_{Random\text{-}Access\text{-}gop}=t_2\text{-}t_{fr}$$

$$\equiv$$

$$t_{Random\text{-}Access\text{-}gop}=(gop-1)*t_{fr}+(log_2(gop)+1)*t_{en\text{-}RA}+ \\ +t_{net}+t_{dec\text{-}RA} \tag{30}$$

Analyzing Eq. (17) and (30) we come to the conclusion that the temporal prediction of the HEVC that will be chosen for a tele-haptic system depends on the available bandwidth of the network, the delay of the network t_{net}, the mean encoding ten and decoding t_{dec} time for each configuration, the frame rate (fps) of the video stream, the data rate that each configuration produces, and the QoS that the specific tele-haptic application requires.

9.4.3 Computational Cost

In order to estimate the level of the encoding time, a personal pc Intel core i3 2100 at 3.1 GHz with 4 GB RAM has been used. For HEVC encoding, the HEVC Test Model HM 16.2 has been used. The video sample was the mobile_cif YUV series [117] with 352×288 resolution at 24 Hz with a duration time of 10 sec. The encoding time and the data rate of the video sample are shown in Table 9-2.

Table 9-2: Computational Cost of HEVC (for 10 sec video)

Inter Prediction	***QP***	***RAM (MB)***	***Data Rate (Kbps)***	***Encoding Time (sec)***	***PSNR***
Intra-Only	32	20.9	3514	341	33.18
Low-Delay	32	47.8	269	1209	31.18
Random Access	32	64.5	290	807	31.66
Intra-Only	27	21.1	5266	374	37.26
Low-Delay	27	47.4	641	1491	34.60
Random Access	27	64.7	584	1012	34.76

From Table 9-2 it is understood that the data rate and computational time of HEVC are strongly dependent on the Quantization Parameter (QP) and the inter prediction configuration. They are inversely proportional. The Low-Delay configuration has much higher encoding time t_{en} than the other configurations. The Intra-Only configuration has the smallest encoding time but the highest data rate. The Random-Access configuration has neutral results regarding encoding time and data rate. The encoding time t_{en} in all the configurations is much higher than the time gap between the successive frames t_{fr}, which means that the real-time encoding is not feasible. The real-time encoding can only be successfully done with the help of parallel processing, a case study that is not in the interest of this chapter.

The Intra-Only configuration has the biggest output of data rate from all the configurations. The Low-Delay configuration outputs a higher data rate than Random-Access. On the other hand the Intra-Only configuration has the smallest encoding time from the other configurations. A more thorough investigation of the HEVC encoding and decoding can be found in [141].

In order to compare the computational cost and the produced data rate of the HEVC with the H.264/AVC, the same video, on the same computer, with the same QP for both encoders were used. The configuration of the H.264 was altered in order to resemble the inter prediction configurations of the HEVC. The outcomes of the H.264/AVC are presented in Table 9-3.

Both encoders have similar PSNR, which means that the quality of the decoded video is similar. The encoding time of the H.264 is, in most of the cases, by far longer than the HEVC's. It can be noticed that the encoding time of the H.264 is not inversely proportional to the factor QP. H.264 uses a lot more RAM from the HEVC. The data rate of the HEVC, in most cases, is smaller than the H.264's. The only case that the data rate of the H.264 is a little smaller than the HEVC is for the random-Access inter-prediction with QP 32. In that case, the encoding time of the H.264 is 1181% longer than the HEVC, and the RAM usage 173% bigger. For the Low-Delay configurations, the data rate of the HEVC is more than 50% smaller than the H.264.

Table 9-3: Computational Cost of H.264/AVC (for 10 sec video)

Inter Prediction	*QP*	*RAM (MB)*	*Data Rate (Kbps)*	*Encoding Time (sec)*	*PSNR*
Intra-Only	32	32.3	3948	1191	33.99
Low-Delay	32	95.9	652	2983	33.37
Random Access	32	111.8	270	9528	32.58
Intra-Only	27	32.3	5974	1416	37.61
Low-Delay	27	80.3	1490	2702	36.70
Random Access	27	112.7	644	6487	35.43

9.4.4 Data Rate Reduction

In the case of a tele-haptic application that is sensitive to the data rate, the Intra-Only configuration should be avoided. In that case, apart from video, the data rate of the haptic stream should be minimized as well. Most of the haptic interfaces [135] produce haptic MU at a rate of 1 KHz. This rather high packet rate is often difficult to transfer through the Internet.

One interesting technique that is proposed for data rate reduction is the dead-reckoning technique [107]. Dead-reckoning can keep the output rate of the haptic media at 1 KHz by prediction and convergence. The haptic source compares the position of the haptic interface to the predicted one. If the difference between the two positions becomes larger than a threshold value, the real position information of the haptic interface is transmitted. This technique has encouraging results in the case of network congestion.

An additional technique, for packet rate reduction, is the packetization interval of the haptic MU [125]. If packetization interval is enforced every P ms on the haptic stream, then the packet rate of the haptic stream can be reduced by a factor of P=8 or P=16, depending on the delay of the system. To achieve information compression, differential coding and quantization is enforced inside the intervals. Each MU has strong correlation with its nearby MU. The packet size of its interval could be reduced from 20+24P bytes to 20+24+3(P-1) bytes. This means that for P=8, the data rate reduction will be at 69.34%. Of course, the packetization interval of the haptic MU adds an extra delay of P ms to the haptic stream. If the delay of the video stream given from Eq. (17) or (23), is longer than the delay of the packetized haptic stream, then a packetization interval could be applied.

9.5 Flowchart of HEVC Encoding for Real-Time Transferring of Video, Audio, and Haptics

If all the variables t_{fr}, t_{en}, t_{net}, t_{dec} are known from the above procedures, then the appropriate inter-prediction configuration can be chosen from Eq. (17) and (23) and the limits of Table 5-1. All the above procedures are integrated in the flowchart of Fig. 9-6.

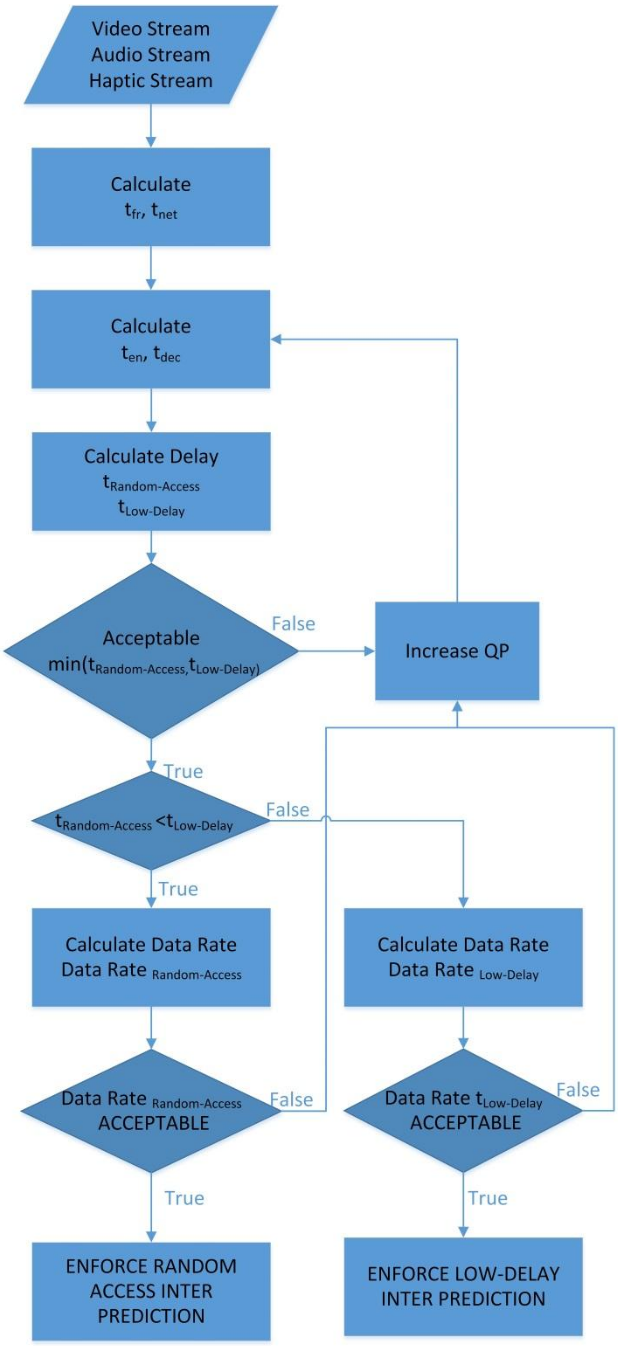

Fig. 9-6: Flowchart of HEVC encoding for real-time transferring of video, audio, and haptics.

10 Real Time Wireless Multisensory Smart Surveillance With 3D - HEVC Streams for IOT

The material in this chapter was presented in [154].

10.1 Introduction

The Internet has revolutionized computer applications and communications. From the simple transfer of text messages we have moved to the Web 2, the Machine to Machine (M2M) communications and the transfer of real-time audio, video and other multisensory data, all together called supermedia. The new evolution stage of the Internet is the Internet-of-Things (IoT). Internet of Things semantically means a world-wide network of interconnected objects uniquely addressable, based on standard communication protocols [155]. Person-to-person, person-to-object and object-to-object communication is now thriving. Humans, sensors, actuators, and smart objects are exchanging vast, real-time information.

One main sector of the IoT is the efficient encoding and transferring of video streams over the Internet. The new High Efficiency Video Encoding (HEVC) offers more than 50% improvement in video compression over its predecessor H.264 Advanced Video Coding standard, with the same image quality, at the expense of the increased computational complexity [115]. The 50% improvement of video compression corresponds to 50% bit rate reduction for video transferring over the IoT. This reduction is crucial, especially in real-time video monitoring for smart surveillance systems. Smart surveillance, is the use of automatic video analysis technologies in video surveillance applications. Smart surveillance systems often deal with real-time monitoring of persistent and transient objects/people within a specific environment. They use image processing algorithms for detection, tracking, and understanding of moving objects/people of interest in dynamic scenes. The main stages of processing in an intelligent visual surveillance system are: moving object detection and recognition, tracking, behavioural analysis and retrieval [156].

The smart surveillance systems are divided in two main categories, depending on where the video analysis is made. The first category includes cameras with processing capabilities. The detection of the event and the storage of the event are made autonomously by the camera. Such cameras are called smart cameras [157]. In the second category, the video stream is transferred through the network in datacenters where the video analysis is made. Large research projects in this category are the Visual Surveillance and Monitoring (VSAM) [158], the Annotated Digital Video for Intelligent Surveillance and Optimized Retrieval (ADVISOR) [159], and the Smart Surveillance System of IBM [160]. In the Third Generation Surveillance Systems (3GSS) [161] a mixture of these categories is made, both smart cameras and large datacenters are being used. The video analysis often includes image enhancement [162], motion detection [163], object tracking [164], and behavior understanding [165].

Smart surveillance systems can be enhanced with the help of 3D monitoring technologies. The 3D camera, also called time-of-flight or depth camera, emits modulated infrared light and measures the time the infrared signal takes to travel from the camera to the object and back again: the elapsed time is called "time of flight".

As we are heading towards the 5th generation of wireless/ mobile broadband networks, numerous devices and networks are interconnected. The Internet of Things (IOT) is becoming a reality. Person-to-person, person-to-object, and object-to-object are continuously exchanging massive real-time supermedia data. The efficient transmission of this data is of great importance for the smart surveillance systems. The scope of this paper is to propose a transport protocol for efficient delivery of supermedia data from smart surveillance systems. Flow/congestion control algorithms and synchronization techniques should be enforced in order to maximize efficiency of a smart surveillance system.

The rest of the chapter is organized as follows: Section 2 presents a high-level system architecture for smart surveillance systems. Section 3 outlines the characteristics of an HEVC – 3D depth video monitoring. Section 4 proposes a synchronization algorithm for intra and inters media synchronization between RGB video, Depth video, audio and other multisensory data. Section 5 presents the wireless network infrastructure of the surveillance system. Section 6 analyzes the proposed protocol for transferring multisensory data streams through the wireless IoT networks. Section 7 presents a case study for transferring a wireless multisensory 3D - HEVC stream over wireless networks. Finally section 8 concludes this paper.

10.2 Smart Surveillance Systems

As the number of threats of burglary, robbery and terrorist activities increases, surveillance systems have become a necessity. The traditional methods for monitoring with the use of CCTV cameras are now evolving to smart surveillance systems. Threat detection and video analysis is automatically made by information systems. Instant notification and alerts produced by smart surveillance systems reinforce public safety and security.

Smart surveillance data collected by cameras and sensors can also provide valuable decision support analysis to organizations. Intelligent analytics enhance business intelligence. The automatic video analytics help organizations to seize opportunities or revise policies.

Smart surveillance systems form a very important sector of video monitoring, video transmission and video analysis. Automated recognition of individuals and objects, pre-determined traits and risks lies at the basis of smart surveillance systems. Some of the innumerable applications of the smart surveillance systems are:

- Unattended surveillance for security reasons
- Automated inspection for quality assurance
- Defect detection and dimensional gauging
- Non contact measurements
- Part sorting and identification
- Code reading and verification
- Robot guidance and automated picking
- Biometric recognition and access control
- Object detection and tracking
- Environment mapping

Apart from video, a smart surveillance system can monitor many more sensory data such as:

- Audio
- Temperature
- Humidity
- Acceleration
- Luminosity
- Pressure
- Chemical Analysis
- “Time of Flight” depth imaging
- Radiation
- Ultrasonic waves
- Motion

All this sensory information is recorded by sensors which are connected to the surveillance system via the net. The Internet of Things is the network which enables these sensors to collect and exchange data through the existing network infrastructure. It allows sensors to be identified and controlled remotely. The IoT improves efficiency and effectiveness. The transport protocols that have already been proposed for the IoT are the MQTT, the XMPP, the COAP, and the 6LOWPAN [166]. Energy consumption, safety and scalability are the main concerns of these protocols.

A proposed high level system architecture of a multisensory smart Surveillance system is depicted in Fig. 10-1.

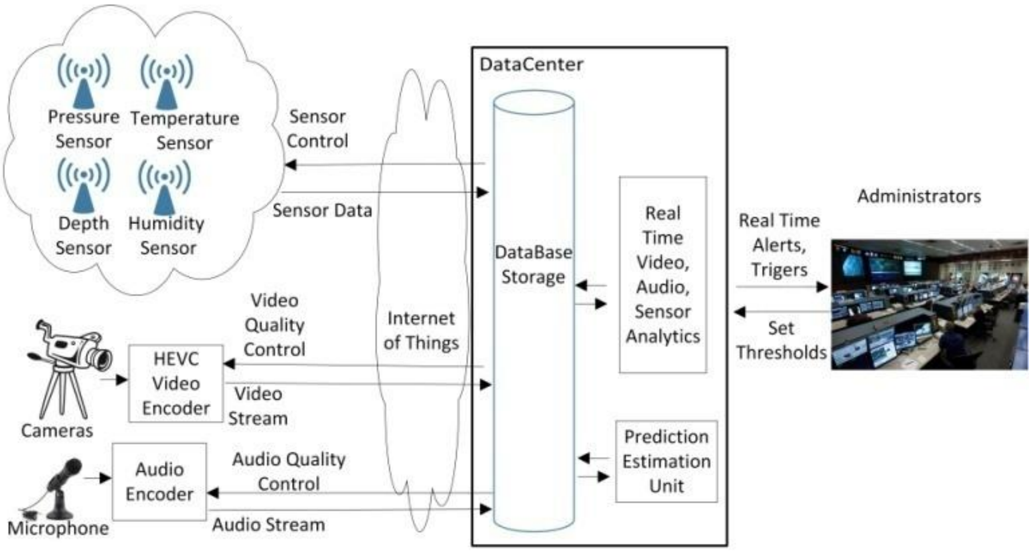

Fig. 10-1: High Level System Architecture of a multisensory smart surveillance system.

In the proposed system architecture three are the main channels that transfer information between the monitored environment and the data center.

The first channel is the **Video Channel**. It transfers visual information from the monitored environment to the data center. In order to lower the bandwidth needed for this transmission an HEVC encoder is used to lower the bitrate, while keeping the image quality as good as possible. Network adaptive congestion control algorithms can be used to absorb all the available bandwidth without forcing the network to congestion [167]. The Real Time Protocol (RTP) is often used to transfer information in this transmission.

Another important channel that is often used in smart surveillance systems is the **Audio Channel**. Advanced Audio Coding (AAC) [168] is usually used for the audio encoding, which is a lossy data compression technique for digital audio. The RTP transport protocol is also used for this channel.

The **Multisensory Channel** is the third data transferring channel. It carries all the sensory data from the monitored environment to the data center. A real time, network adaptive, proposed by the authors, protocol is used for the transport of this information. It is often a channel with low bandwidth requirements but with high update rate frequencies. Moreover, algorithms that enhance the reliable transfer of data should be applied.

The **Data Center** stores all the information from the remote monitored environment. It is often located away from the monitored environment at a secure location, connected to the Internet. All the data stored there is analyzed and real time alerts are triggered, based on the thresholds set by the surveillance administrators. The data center also includes a Prediction and Estimation Unit (PEU). Its main functionality is to perform prediction algorithms on historical database records. Future potential risks can be avoided with the PEU. Data mining algorithms for statistical model building are also applicable.

10.3 High Efficiency Video Coding (HEVC) With 3D Features

10.3.1 HEVC Video Encoding

As mentioned in the previous chapter, the HEVC is ideal for real-time video transferring through the Internet. It achieves more than 50 % improvement in video compression and bitrate over the existing H.264 Advanced Video Coding standard, for the same image quality [168]. The 50% reduction of the bitrate together with the improvement of Internet network conditions made the real time streaming of high and ultra-high definition video over the Internet feasible.

As smart video surveillance focuses on real time video transmission, the encoding time and the data rate are very important factors for the selection of the inter prediction mode. Between the Intra-only and the Low-delay mode of the HEVC encoder, the second is preferred as it presents much lower data rates. Between the Low-Delay and the Random-Access mode, again the Low-Delay mode is preferred because the encoder doesn’t have to wait for the whole GoP to appear in order to start the encoding process [118].

Comparing Table 9-2 and Table 9-3 to the Low Delay mode, it is obvious that the data rate with the HEVC encoder is reduced by 58.74 % for Quantization Parameter (QP)

=32 and 56.91% for QP=27. This reduction indicates HEVC as a perfect encoder for real time video transmission.

10.3.2 Depth Video

Apart from the ordinary video, another interesting factor that can be used in the smart video surveillance is the 3D-Depth camera. These cameras measure the distance between the depth camera and an object with the help of modulated infrared light. Given that the speed of light is known, the "time of flight" that takes the light to travel from the camera to the scattered object and return back reveals the distance between the camera and the object. If this distance is measured for every pixel of the video, the ordinary video is transformed to a 3D-depth video. The depth video can enhance smart surveillance at motion surveillance, object tracking, acceleration and speed monitoring, and 3D-environment mapping. “Time of Flight” monitoring is invariant to visible lighting and weather conditions. The first widely used depth camera was the Microsoft Kinect camera [169] for the Xbox. It produces an 11 bit 640 × 480 depth image at 30 frames per sec. Its precision is inversely proportional to the distance, and decreases from 1 cm at two meters to 10 cm at six meters. If no compression technique is enforced, the data rate for an 11 bit 640 × 480 depth image at 30 fps is 3379,2 Kbps. For a real time transmission over the Internet it is quite a high data rate, which means that intra and inter compression techniques should be enforced on the depth stream for the limitation of the data rate.

A sample of an RGB-Depth image is depicted at the right side of Fig. 10-2. The depth image, in the middle was produced by a Microsoft Kinect camera [170]. It is obvious that most of the objects of the left RGB picture can easily be distinguished and surveilled in the processed RGB-D image.

Fig. 10-2: Output from the RGB camera (left), preprocessed depth (center) and processed RGB (right) for the image [170] .

10.4 Synchronization of HEVC Streams With Depth Video and Multisensory Data

The video camera, the depth camera, the audio and all the other sensors produce frames at a different rate, which means that these frames are loosely coupled. In order for these frames to be synchronized, a synchronization algorithm should be enforced. The proposed synchronization algorithm is the enhanced Virtual-Time Rendering (VTR) media synchronization algorithm [151].

Enhanced VTR enforces both intra and inter synchronization in all streams. The intra synchronization tries to keep the outputs media Units (MUs) to the destination, at the same intervals as the generation ones at the source. The inter synchronization reconstructs the original temporal relations between the MUs of different streams.

In order for the enhanced VTR to be enforced, one of the multisensory streams should be defined as the master stream. As the smart surveillance mostly depends on the visual sense, the video stream is often selected as the master stream.

10.4.1 Intra Synchronization

The VTR algorithm enforces the intra synchronization for each stream separately. Each Media Unit (MU) that is sent from the source should be an output at the destination at an ideal output time.

The process of the VTR algorithm for intra synchronization was fully analyzed in chapter 9.4.1 .

10.4.2 Inter Synchronization

As soon as the intra synchronization for all the data streams is started, the inter-synchronization among all the streams should be enforced. The inter-synchronization will be based on the new scheduled output time *dn* for each stream.

As mentioned earlier, the master stream is often the video stream. Let us assume that the scheduled output time of the video stream is the dv_n, and the scheduled output time of the audio, the depth and other sensors stream is da_n.

All the first packets from each stream should get the same output time, which means:

$$dv_1=da_1 \tag{31}$$

This means that based on Eq. (13), the first ideal output time for the video xv_1 and the other sensors xa_1 should be:

$$xv_1=xa_1 \tag{32}$$

As the video stream was set as the master stream, its ideal output time should not be changed. This means that if xv_1 is bigger than xa_1, the secondary stream should be delayed and have as final output time xa_1*

$$xa_1*=xv_1 \tag{33}$$

If the video stream has smaller ideal output time for its first frame xv1 than the other streams, the video stream should not be delayed. The other streams should get ideal output time for its first frame

$$xa_1^* = \begin{cases} Aa_1, & if\ Aa_1 \geq xv_1 \\ xv_1, & otherwise \end{cases} \tag{34}$$

where Aa_1 is the arrival time of the first sensor frame.

All the modified output time for the other frames is based on Eq. (14), (15), (16).

10.5 Wireless Network Infrastructure

All the data from the remote surveilled environment to the data center are transferred wirelessly. All wireless communication standards that could be used for this transmission are shown in Table 10-1.

Table 10-1: Wireless Communication Standards

	Max Range (m)	***Max. Upload Data Rate (MBits)***	***Max Power Consumption (mW)***	***Frequency (MHz)***
ZigBee [171]	70	0.25	30	2400
Bluetooth [171]	100	1	100	2400
802.11ac [172]	150	1300	1000	5000
4G [173]	Cellular based	75	200	900/ 1800/ 2300
5G [174]	Cellular based	1 GBps	Lower than 4G	Undifined

The most promising, but not yet available, wireless standard is the 5G (5th generation mobile networks or 5th generation wireless systems). Stakeholders claim it will be available by 2020. Its target is to provide bandwidth greater than 1 GBps, end to end delay smaller than 1 ms, and efficiency higher than its predecessor 4G .

The prevailed and available standards of Table 10-1 are the 4G and the 802.11ac because of the available range and data rate that they offer. The 4G uses the cellular network to offer unlimited range. On the other hand, the range of a Wi-Fi network may be extended to only some several hundred of meters but is a lot faster than the 4G. If the distance between the surveilled environment and the access point of an available WiFi is some dozens of meters then the 802.11ac standard is preferred. If the surveilled environment is at a distant location, then the 4G standard is used.

10.6 The Proposed Protocol for Multisensory Surveillance System.

In this chapter a new network Adaptive Multisensory Real-time Transmission Protocol (AMRTP) is proposed. Its main target is to reliably send, real-time multisensory information from the remote surveilled environment to the data center. In order for the protocol to be network adaptive, the delay time and the packet loss are recorded at the receiver. The protocol changes its sending rate and packet size according to the network conditions. Its primary target is to send all the data reliably without forcing the network to congestion.

10.6.1 Adaptive Packet Frame Grouping

One method that the AMRTP uses to avoid congestion is the adaptive packet frame grouping [175]. When network conditions deteriorate, the protocol lowers its sending rate in order to avoid congestion. Given that a sensor usually produces data packets at a steady rate, the protocol should group these packets in bigger frames in order to lower the sending rate. The number of grouped packets is changed according to the network conditions. When network conditions deteriorate, the number of the grouped packets increases, in order for the sending rate to be reduced.

The maximum number of grouped packets is dependent on the update rate of the sensor, the minimum acceptable refresh rate at the receiver and the maximum acceptable delay that a service can tolerate in order not to affect its real time feature. Based on the Mean Opinion Score (MOS) on the Quality of Experience (QoE) measurements [5], the Quality of Service (QoS) requirements for real time services is depicted in Table 5-1. As most sensor data are illustrated through graphics, the QoS requirements for the AMRTP protocol are depicted in the fourth column of Table 5-1.

If the sensor update rate is SUR packets per second, and the maximum affordable delay is 300 ms, the maximum number of grouped packets np_{max} per frame is:

$$np_{max}=0.3*SUR<512 \tag{35}$$

which correspond to the minimum sending rate of 33.33 Hz.

The minimum number of grouped packets is set to

$$np_{min}=SUR/100 \tag{36}$$

packets per frame, which correspond to a sending rate of 100 Hz, which is a perfect refresh rate for the visual sense.

The number of the grouped packets np changes according to the Eq. (37). The delay of the network d_{net} can easily be extracted if ICMP packets are sent from the sender to the receiver every $T_{ping}=0.5$ sec. The factor d_{max} is the maximum acceptable network delay. For real time graphics the maximum acceptable delay is set to $d_{max}=300$ ms.

$$\mathrm{np_i} = \begin{cases} \mathrm{np_0} = \mathrm{np_{min}} & \\ \mathrm{np_{i-1}} - 2 > \mathrm{np_{min}} & ,\mathrm{dnet} < dmax/3 \\ \mathrm{np_{i-1}} - 1 > \mathrm{np_{min}} & ,\mathrm{dmax}/3 < dnet < 2 * dmax/3 \\ \mathrm{np_{i-1}} & ,2\mathrm{dmax}/3 < dnet < dmax \\ \mathrm{np_{i-1}} + 2 < \mathrm{np_{max}} & ,\mathrm{dmax} < dnet \end{cases} \tag{37}$$

The delay of the network has been divided into four intervals. The first delay interval, $\mathrm{dnet} < dmax/3$, corresponds to perfect network conditions. The protocol tries to increase the sending rate rapidly. The second interval, $\mathrm{dmax}/3 < dnet < 2 * dmax/3$, corresponds to good network conditions and the protocol tries to increase the sending rate slowly. The third interval, $2 * \mathrm{dmax}/3 < dnet < dmax$, corresponds to acceptable network conditions and the protocol tries to keep the sending rate steady. The last interval $\mathrm{dmax} < dnet$ corresponds to unacceptable network conditions and the protocol tries to lower the sending rate in order to avoid congestion.

10.6.2 Adaptive Quantization

Another method that the AMRTP uses to avoid congestion is the network adaptive quantization [176]. The quantization levels of the data values change according to the network conditions. When the network conditions are good, the quantization levels increase and vice versa. The maximum quantization level is set to $ql_{max} = 65536 = 2^{16}$, which demand 2 bytes per data value. The minimum quantization level is set to $ql_{min}=256 = 2^8$ which demand 1 byte per data value. An intermediate value of quantization level is set to $ql_{medium} = 4096 = 2^{12}$, which demands 1.5 bytes per data value. The quantization level ql changes according to the Eq. 38.

$$\mathrm{ql} = \begin{cases} 65536 & , \mathrm{dnet} < dmax \\ 4096 & , \mathrm{dnet} > dmax \ (For\ t > \mathrm{T_{maxdelay}}\ ) \\ 256 & , \mathrm{dnet} > dmax \ \ (For\ t > 2 * \mathrm{T_{maxdelay}}\ ) \end{cases} \tag{38}$$

If the delay of the network is bigger than the maximum acceptable delay d_{max} for a period of time $T_{maxdelay}=2.5$ sec, then the quantization level is degraded to 4096. If the delay of the network remains bigger than d_{max} for a period of time greater to $2*\ T_{maxdelay}$, then the quantization level is degraded to 256.

10.6.3 AMRTP Packet Header

The AMRTP protocol runs over the UDP protocol. The UDP protocol is chosen as it is the lightest and faster transport protocol. The AMRTP is used to compensate the UDP's lack of reliability and network adaptation.

The Header of the AMRTP protocol for one packet is illustrated in Fig. 10-3.

Bits	***0 - 15***	***16-17***	***18-22***	***23-31***
0	Sequence Number	QuanL.	SenId	NumPack.
32	Data			

Fig. 10-3: Header of AMRTP protocol

The Sequence Number is used to reinforce the reliability of the protocol through the acknowledgment process.

The QuanL. (Quantization Level) field informs the receiver of the quantization level of the data and the length of each data value.

The SenId (Sensor ID) field informs the receiver of which sensor the data corresponds to.

The NumPack (Number of Packets) field informs the receiver of the number of packets that are grouped in this frame. As the NumPack consists of 9 bits, the maximum number of grouped packets is 512.

10.6.4 AMRTP Reliability

AMRTP protocol, apart from being efficient and network adaptive, should also be reliable. As a real time protocol, it should timely deliver the packets. But as the packets include surveillance information that could be used for post processing, all the packets should be reliably transferred.

In order to enforce reliability, a sequence number is used. Every frame has a unique sequence number. The method of the cumulative negative acknowledgement (CNAK) [177] is applied. The sender sends the frames as soon as they are created. A copy of these frames is stored in a buffer at the sender side in case a negative acknowledgement is received. A CNAK is sent by the receiver every k= 1 sec containing the frames that have not been received. The receiver also sends the last sequence number that has been received to help the sender to empty its buffer.

The packet construction for the cumulative negative acknowledgment is depicted in Fig. 10-4.

<table>
<tr><th>Bits</th><th>0-3</th><th>4-7</th><th>8 - 15</th><th>16-31</th></tr>
<tr><td>0</td><td>SensorId</td><td colspan="2">ACK Sequence Number</td><td>Last Received Sequence Number</td></tr>
<tr><td>32</td><td colspan="4">Data (Dropped Sequence Numbers)</td></tr>
</table>

Fig. 10-4: Header of AMRTP CNAK

This CNAK packet should be sent reliably. An acknowledgment over the UDP protocol, containing the lost frames, is sent by the sender as soon as it receives a CNAK. If the receiver doesn't get this acknowledgment in a specific time period $h= 1$ sec, a CNAK is resent.

The flow diagram of the AMRTP protocol is depicted in Fig. 10-5.

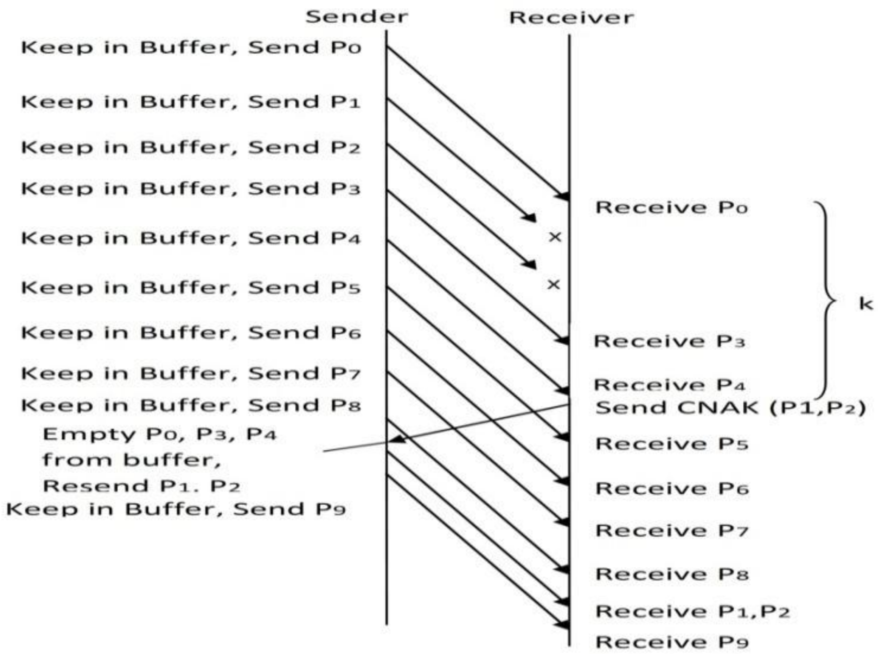

Fig. 10-5: Flow Diagram of AMRTP Protocol

10.7 Measurements for Real Time Transferring of 3D HEVC Multisensory Streams.

The AMRTP protocol was tested in two different network topologies, with two different sensors and two different network traffic scenarios. In all cases, the performance of the AMRTP protocol was compared against the fast but unreliable UDP with no adaptive frame grouping and quantization. The performance metrics that were measured were the packet loss, the average end to end delay, and the standard variation of the delay, also known as jitter.

In the first topology, the surveillance sensor sent its data through a 802.11n WiFi 300 Mbps wireless network. In the second topology, the surveillance sensor sent its data through a 3G HSDPA network.

Two different sensors were used. The first sensor had an update rate of 30 packets per second, which is usually encountered in depth cameras. The second sensor was producing packets at a rather high update rate of 1000 Hz. The value size of each sensor packet was changing based on Eq. (38) from 1 to 2 bytes according to network conditions. The packet frame grouping was changing based on Eq. (35)-(37).

Two different network traffic scenarios were used. In the first case, no other network traffic was sent over the wireless/ 3G network. In the second case a HEVC 1080p at 30 fps video and a AAC-LC with 96 khz sample rate audio stream was sent simultaneously with the sensor stream over the RTSP protocol.

Three measurements for each case of the above transport protocols, network topologies, sensors and network traffic scenarios have been taken. For each measurement 10000 frames were sent. The best results from each of the above measurements are depicted in Table 10-2 and Table 10-3.

Table 10-2: AMRTP and UDP transmission over WiFi and 3G network with no other network traffic.

TOPOLOGY	*PROTOCOL*	*SENSOR UPDATE RATE Packet/sec*	*AVG DELAY ms*	*JITTER ms*	*PACKET LOSS %*
WiFi	AMRTP	1000	2.36	1.355	0.15
WiFi	AMRTP	30	4.45	10.58	0
WiFi	UDP	1000	3.74	22.89	0.27
WiFi	UDP	30	4.49	12.74	0.1
3G	AMRTP	1000	70.51	97.64	0.15
3G	AMRTP	30	60.29	88.1	0.99
3G	UDP	1000	248.96	261.58	29.74
3G	UDP	30	57.16	35.83	0

Table 10-3: AMRTP and UDP transmission over WiFi and 3G network simultaneously with a 1080p video stream.

TOPOLOGY	*PROTOCOL*	*SENSOR UPDATE RATE Packet/sec*	*AVG DELAY ms*	*JITTER ms*	*PACKET LOSS %*
WiFi	AMRTP	1000	2.88	3.3	0
WiFi	AMRTP	30	3.19	3.58	0
WiFi	UDP	1000	2.85	2.16	0
WiFi	UDP	30	2.50	2.7	0
3G	AMRTP	1000	172.31	323.29	0
3G	AMRTP	30	114.02	213.60	0
3G	UDP	1000	570.67	402.39	26.66
3G	UDP	30	202.89	332.11	0

It is understood that in both network traffic scenarios WiFi network conditions are by far better than the 3G network. The average delay and jitter are smaller to the WiFi network.

Average end to end delay is almost invariant to the enforced transport protocol, to the sensor update rate and the traffic network scenario in the case of WiFi network topology.

In the case of the 3G network, the average delay increases when the sensor update rate is increased.

The packet loss in most circumstances is below 1%. The only scenario that shows increased packet loss is the UDP transport protocol over the 3G network with sensor update rate equal to 1000 packets per sec. The packet loss in these circumstances is 26-29 %, which is unacceptably high. This high packet loss can be lowered if the sending rate is reduced. When the UDP transport protocol is used, and the sending rate changes from 1000 packets per second to 30, Table 10-2 and Table 10-3 depict that the packet loss is reduced from 26.66 and 29.74% to 0%. This means that the 3G network is not capable to transfer 1000 packets per second, and many packets are dropped. This observation is being exploited by the AMRTP protocol and reduces the sending rate by grouping the packets to frames. That's why the packet loss in the case of the AMRTP is so small.

As the packet loss is almost zero in the case of the AMRTP protocol, the cumulative negative acknowledgment is the best reliability mechanism, as it reduces network traffic from the unnecessary acknowledgments. All the packets that are lost in the AMRTP protocol are resent successfully with the reliability mechanism.

Comparing the AMRTP to the UDP transport protocol, it is understood that they have almost the same results as in the case of the WiFi network. The advantage of the AMRTP protocol is its reliability, due to the cumulative negative acknowledgement that it enforces.

In the 3G network, the AMRTP protocol outperforms the UDP protocol in almost all the performance metrics. As the network conditions of the 3G network are inadequate for the transmission of high update rates, the adaptive packet frame grouping and the adaptive quantization of the AMRTP protocol enhance the performance of the transport protocol.

Apart from the main performance metrics, delay, jitter and packet loss, the adaptive behavior of the frame grouping and the quantization was monitored.

In the case of the WiFi network topology, the network conditions were rather adequate for the multisensory streams. The number of the grouped packets per frame was constantly equal to the minimum number of grouped packets np_{min} based on Eq. (36). Similarly, the quantization level of Eq. (38) remained equal to its maximum level in all WiFi experiments.

On the other hand, the 3G network with video traffic was rather congested. The number of the grouped packets and the packet size were fluctuating according to the network conditions, in order to avoid congestion.

Fig. 10-6 depicts the adaptive number of grouped packets per frame for the 3G network with and without network traffic.

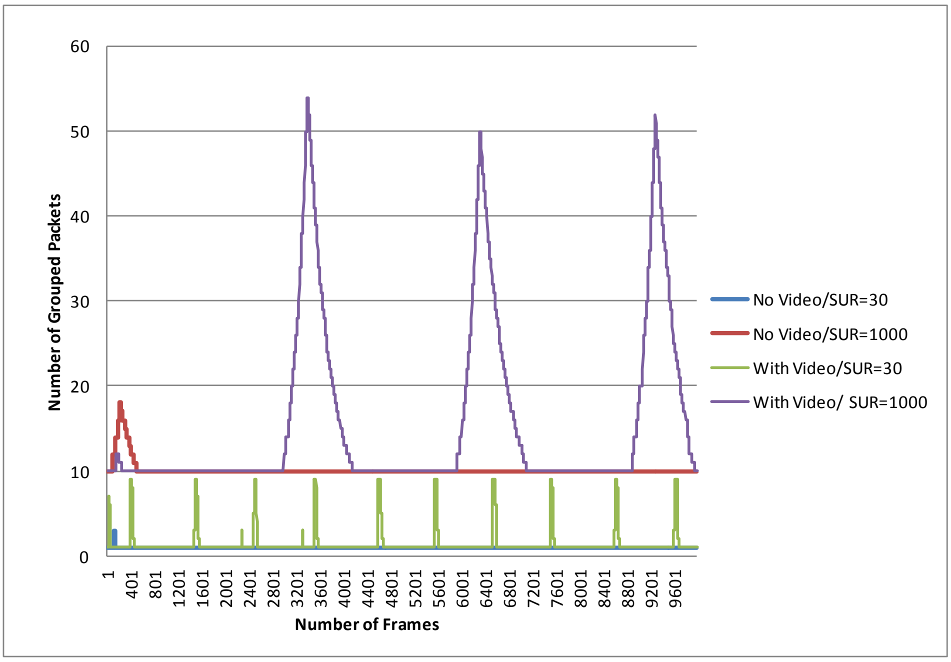

Fig. 10-6: Adaptive number of grouped packets per frame

Based on Eq. (36), when the Sensor Update Rate (SUR) is equal to 30, the minimum number of grouped packets is 1. When the SUR is equal to 1000 the minimum number of grouped packets is 10. It is obvious that when video traffic is enforced, the network is rather congested and the number of grouped packets often increases.

Fig. 10-7 depicts the adaptive quantization level for the WiFi and the 3G network with and without network traffic.

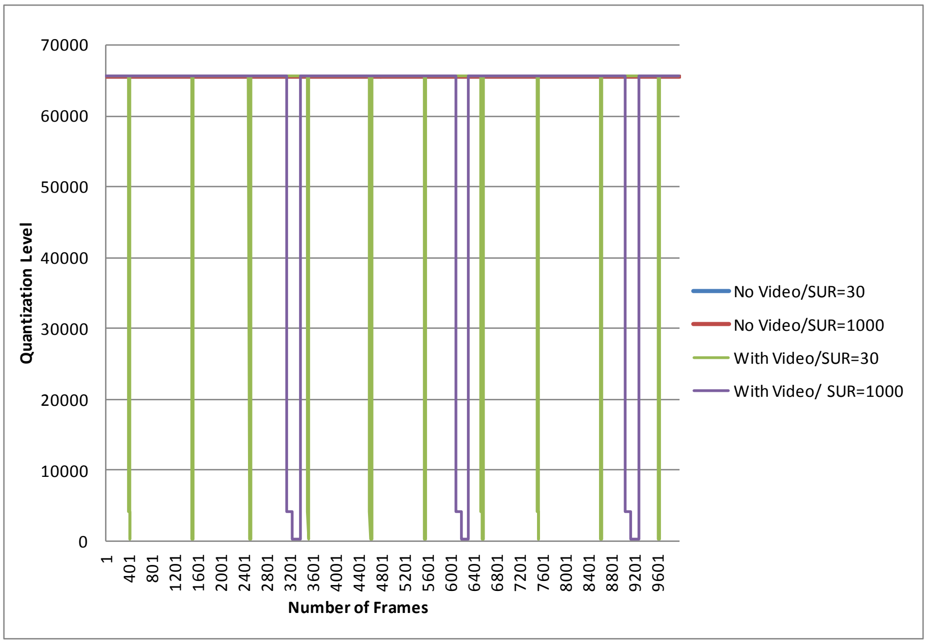

Fig. 10-7: Adaptive quantization level

When the 3G Network doesn't transfer any video traffic, the quantization level is constant at its maximum value 65536. When the 3G network transfers video traffic the quantization level fluctuates from 65536 to 256, in order to lower the packet size, and avoid congestion and packet loss.

11 Transferring Wireless High Update Rate Supermedia Streams Over IoT

The material in this chapter was presented in [178].

11.1 Introduction

Over the last years the Internet has evolved from the WWW, the simple transfer of text and static pages to the Web 2, the M2M communication, and the transfer of real-time supermedia. The new stage of Internet evolution is the Internet of Things. The IoT improves efficiency, effectiveness and offers new business opportunities. Apart from person-to-person communication, the person-to-object and object-to-object communication is now evolving. Apart from humans, sensors, actuators and smart objects are now exchanging information. IoT devices have exceeded 30 billion. The number of the message transactions is increasing rapidly. One of the challenges that the IoT has to overcome is to ensure the Quality of Service (QoS) for all participants and maximize the Quality of Experience for all users. New protocols such as the MQTT, the XMPP, the COAP, and the 6LOWPAN have already been proposed [166]. These protocols take into consideration issues such as energy consumption, safety and scalability. The main objective of IoT should be the Quality of Experience of the user. As Internet network conditions are constantly changing, transport protocols and flow control algorithms should monitor network conditions and modify the supermedia streams of data.

In this massive network, wireless communications are invited to play a major role. Most of the end users of IoT transfer their data wirelessly. This gives them the ability to move freely with no wire constraints. All wireless communication standards that could be used for this transmission are shown in Table 10-1.

One major part of the data that will be transferred over the IoT regards to sensors and actuators. An important segment of this data refers to the haptic human sense. Haptics is the science of manual sensing and manipulation of objects through the sense of touch. Haptic data play a major role in the Quality of Experience of the user, transform virtual reality to augment reality and evolve multimedia to supermedia. The main obstacle that impedes tele-haptics from flourishing is the delay and the jitter that is encountered in the wireless IoT. Until 5G is available to the public, flow control algorithms should be defined in order to mitigate congestion, latency and packet loss in IoT under the available wireless communications.

11.2 Flow Control Filtering Algorithm for Supermedia Transferring Over IoT

Supermedia streams demand different QoS for each media stream. In order to fulfill these QoS, different flow control techniques should be enforced in each stream. Streams with strict QoS should have higher priority than others. Moreover, since the network conditions of the IoT are time-varying, the flow control algorithms should be network adaptive. The transmission rate, the packet size and the throughput of each stream should be network adaptive. If the network is showing signs of congestion, the transmission rate and the packet size should be reduced in order for heavy congestion to be avoided. If network conditions are deteriorating, packets with lower priority should be filtered and dropped. Adaptive differential coding and quantization should modify the packet size according to the network conditions.

The proposed network adaptive flow control algorithm for supermedia streams over the IoT is depicted in Fig. 11-1. Each step of the algorithm is analyzed on the following subsections.

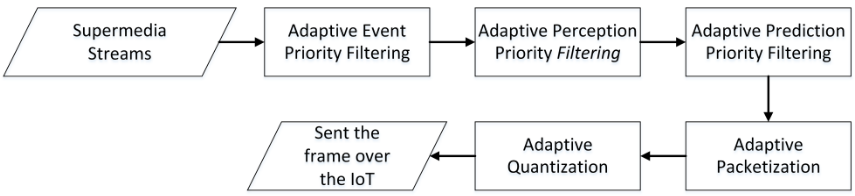

Fig. 11-1: System model of network adaptive flow control algorithm for supermedia streams over IoT.

11.2.1 Network Adaptive Event Priority Filtering

Packets that correspond to significant events/keypoints should have higher priority than other simple update packets and should be sent more reliably [127]. When network conditions deteriorate, packets with lower priority should be buffered or dropped. Such a keypoint could be an Instantaneous Decoder Refresh (IDR) frame of a H.264/265 video stream, or a significant haptic event, such as grasping a virtual object.

11.2.2 Network Adaptive Perception priority

The perception priority is based on the Weber's law of Just Noticeable Difference (JND) [133]. Eq. (39) calculates the JND. The factor I is the stimulus intense, that the supermedia interface causes to the user. The dead-reckoning theory [128] supports that supermedia packets that generate small perceptible feelings ΔI to the user should be dropped. The constant κ is called the Weber fraction.

$$\Delta I = I * \kappa \tag{39}$$

When the supermedia packets produce to the user difference on the stimulus intense dI higher than the threshold ΔI, then the packet should obtain high priority. The higher the dI is, the higher the priority should be.

In order for the algorithm to be network adaptive, the Weber fraction κ should change according to the network conditions. The constant κ should increase when network conditions deteriorate, in order to lower the throughput of the stream. The bigger the Weber fraction is, the smaller the QoE of the user.

11.2.3 Network Adaptive Prediction priority

Packets that can be predicted by previous packets should obtain a lower Prediction Priority [126]. The prediction unit is installed both at the sender and at the receiver. If the prediction unit at the sender calculates that the current packet could be predicted at the receiver from the last packets that were sent, then the current packet is not transmitted. In order for the algorithm to be network adaptive, apart from the identical predicted packets, predicted packets that are similar to the real packets could be excluded from the transmission. Again this algorithm should be based on Weber's law of the JND to decide which packets could successfully be predicted at the receiver side. If the predicted packet does not produce greater difference on the stimulus intense ***dI*** to the users from the real packet than the threshold $\Delta I'$, then the packet is not transmitted, but it will be predicted on the receiver's side.

11.2.4 Network Adaptive Transmission Rate

The source should adapt its transmission rate according to the network conditions [57]. If the supermedia interface produces update packets steadily and the sender fluctuates the sending rate, a buffer is necessary at the sender side to absorb the fluctuation. The negative aspect of this technique is that if the supermedia interface produces packets at a very high update rate, for haptics this is usually 1 kHz, the sender should transmit its packets sometimes even faster to compensate the previous lower rates. Such a high update rate often results in network congestion and packet loss. In order to lower the update rate, an interesting proposal is to integrate a group of packets in a frame and send them as a unified packet, a technique called packetization interval. Fujimoto and Ishibashi [125] have proven that a packetization interval with 8 packets per frame that is sent every 8 ms instead of 1 packet/sec improves system performance in overloaded networks. This means that the sending rate could be lowered to the 1/8 of the 1 kHz.

11.2.5 Network Adaptive Quantization and differential coding.

The bandwidth that a supermedia stream absorbs depends on two factors, the frame rate and the size of the frame. The frame size could be reduced if differential coding and quantization [132] is enforced on the packets that are grouped in the frame. The technique that is recommended is the Differential Pulse-Code Modulation (DPCM). The differential coding transmits not the whole packet but the difference between the reference point and the processed packet. Differential coding produces smaller packets than the original ones. Smaller packets mean fewer bits. The quantization of the differentiate values could be made with variable quantization step. The Adaptive Differential Pulse-Code Modulation (ADPCM) for haptic packets was introduced in [132]. The quantization step should be changed according to the network conditions. When the network conditions deteriorate, the quantization step should increase in order for fewer bits to be required for the reconstruction of the original values.

11.3 Measuring Performance For High Update Rate Streams over WiFi Repeaters.

Network conditions refer to the amount of traffic that is being transferred through the Internet, the End to End delay, the jitter between source and destination, and the available bandwidth for data transport.

All the above metrics vary in time and space. They depend on the number of the online users, the amount of data that is being transferred at the specific moment of the measurement, and the available equipment of lines and routers. The amount of data transferred through the web is constantly increasing. The number of the online users is increasing as well. The growth of data transfer is compensated by continuing infrastructure upgrades of computer networks.

There are two types of approaches for monitoring the network status. The two disciplines of network monitoring are the active and the passive measurements [109]. In the active measurement, specific generated probe packets, ICMP messages, are sent to specific destinations; measurements for delay, round trip time, jitter and packet loss are made. Some common diagnostic tools for active measurements are the ping, traceroute, capprobe, pathchar, netem and dummynet [110]. On the other hand, passive approach is based on the observation of the traffic that flows on the links. Some passive monitoring tools, commonly called sniffers, are the Tcpdump, Wireshark, Ethereal, Netflow and JFlow [109].

The authors actively measured the average, the standard deviation of the delay, and the packet loss rate in a WiFi multihop network with wireless access point repeaters. Two different topologies were used for these measurements. The first scenario was a simple WiFi network with one Access Point (AP). The second scenario was a WiFi network with one wired AP and one wireless AP repeater. The access points that were used were the 300 Mbps Tenda A30 Wireless N300 Range Extender. The packet size and the update rate of the streams were changing in order to examine whether the transferring of high update rate of supermedia streams through wireless multihop networks is possible. Two different update rates were used in all network topologies, a stream with 1000 packets per sec and a stream with 500 packets per sec. The packet size of the stream was changed from 64 bytes to 128 and 256 bytes. The results of these measurements are shown in Table 11-1.

Table 11-1: WiFi Network Status with/without AP Repeaters

Network	***Update Rate (Packets /sec)***	***Packet Size (Bytes)***	***Round Trip Time (ms)***	***Standard Variation of RTT (ms)***	***Packet Loss (%)***
WiFi with no AP Repeaters	1000	64	3.65	6.19	0.06
WiFi with no AP Repeaters	1000	128	3.8	6.59	0
WiFi with no AP Repeaters	500	128	3.21	5.15	0
WiFi with no AP Repeaters	500	256	3.78	6.18	0.07
WiFi with 1 AP Repeater	1000	64	11.18	8.85	0.39
WiFi with 1 AP Repeater	1000	128	14.97	17.98	0.8
WiFi with 1 AP Repeater	500	128	7.37	8.20	0.4
WiFi with 1 AP Repeater	500	256	10.98	15.16	0.42

It is obvious that even the round-trip delay of Table 11-1 is by far smaller than the QoS requirements regarding the delay of Table 5-1. On the other hand, the jitter of Table 11-1 is bigger than the jitter QoS requirement of Table 5-1. This exceedance can be limited if a network adaptive buffer is placed at the receiver side to absorb the delay fluctuation and diminish the jitter.

Furthermore, it is understood that when the update rate increases and the packet size remains steady, the network delay, the jitter and the packet loss are increased as well. This means that the network conditions deteriorate. Similarly, when the packet size is increased and the update rate remains the same, the network conditions deteriorate as well.

An interesting case is when the update rate is minimized to half, from 1000 to 500 packets/sec, and the packet size doubles its size, from 128 bytes to 256 bytes. This means that the throughput of the application remains the same. The network conditions in this case are improved. This improvement is more obvious when the wireless network is consisted from wireless AP repeaters. The proposed Flow Control Filtering Algorithm of Fig. 11-1 takes advantage of this observation and tries to lower the sending rate of the stream by grouping the packets into bigger frames, especially when network conditions are not adequate. Moreover, compression techniques such as differential coding and quantization are enforced to these frames in order to minimize the transferred frames. From Table 11-1 it is understood that lower sending rates and smaller transferred packets minimize delay, jitter and packet loss.

12 Conclusions

12.1 Summary of the contributions

Parts of this thesis have appeared in varying forms of conferences and journal papers (see appendix A). Particularly, this thesis described the architecture of a Haptic System in [2]. It analyzed transmission characteristics and the flow requirements of Haptic data [2], [4]. It presents the protocols that are used for the transmission of Haptic data and analyzes the qualitative features that these protocols should fulfill [4]. The network conditions of the Internet for different networks, geographical position and time were measured [3], [108].

Moreover it evaluates and compares the most known real-time protocols in a simulation environment [3]. To evaluate the behavior of Haptic protocols, specific performance metrics were attained. Measures of the performance of a Haptics transport protocol relied on tangible attributes. That is, Throughput or Goodput at the receiver end, packet loss, network delay, and Packet Delay Variation also called jitter.

The simulation tests that have been carried out in [3] revealed which protocols could be used for the transfer of real-time supermedia data.The simulation produced accurate statistical and qualitative data that allowed in-depth evaluation of the reported protocols. Taking the above information into consideration, it is possible to suggest corrections to the relevant protocols or create a new haptic protocol.

This thesis also proposed an efficient algorithm for transferring a real-time HEVC stream with haptic data through the Internet [118]. The H.264 and the HEVC are compared. The transferring delay of all the inter prediction configurations of the HEVC were defined.

If the QoS of the network, the encoding and decoding time of the HEVC are known, then the correct temporal prediction of the HEVC could be chosen. It has been proven that the encoding order of the frames inside a GOP could play a major role in the delay of the system. Comparative tests between H.264 and HEVC have been undertaken.

The synchronization techniques that are proposed in [118] compensate the barrier of the low tolerance that haptic data have to jitter and delay.

Also a network adaptive flow control algorithm named NAFCAH is presented [121]. It is quite a flexible algorithm where the user can set its sensitivity to the network variations. All the known congestion and flow control techniques have been enforced, in order to achieve the desired result. Packet priorities such as the event priority, the perception priority and the prediction priority are described and defined. Packetization Interval technique and lossy compression methods such as the Adaptive Differential Pulse-Code Modulation are enforced.

This thesis also describes the architecture of a real time, wireless, multisensory smart surveillance system [154]. All the possible data streams are presented. The data center is proposed as a repository for data storage, as well as a data processing and prediction unit. Real time alerts produced and future threats are predicted. The advantages of a HEVC video stream over its predecessor H.264 for a surveillance system are presented. 3D monitoring techniques are proposed. Intra and inter synchronization techniques for real time multisensory streams are analyzed and enforced. All the possible wireless standards are depicted. The advantages of the 802.11ac/n and the 4G networks are analyzed.

In [154], a novel, reliable, network adaptive transmission protocol, named AMRTP, is also proposed. It successfully adapts its sending rate and bandwidth through the adaptive packet frame grouping and adaptive quantization in order to avoid congestion and packet loss. Experimental tests have shown that the proposed protocol is a promising candidate for real-time multisensory surveillance systems over Wi-Fi or 3G networks. The reliability and the congestion control mechanism that it enforces, sets the AMRTP protocol as a good solution for the reliable transferring of data over time varying network conditions.

The experiment tests that have been carried out in [108] have proven that the transfer of real time high update rate supermedia streams over IoT wireless multihop networks is challenging. In wired networks the network conditions are sufficient for high update rate transferring. In wireless multihop networks high update rates cause increased end-to-end delay, jitter and packet loss. This increase is observed especially when the update rate is high, or when the transferred packet size is getting bigger or when the hops in the wireless network are multiplied. In order to minimize these effects, a flow control algorithm for high update rate supermedia data transferring over wireless multihop networks is proposed in this paper. Prediction, compression, packet priority and filtering techniques are enforced in order to minimize the update rate and the packet size of the supermedia streams. Applying the proposed flow control delay, jitter and packet loss are reduced.

Based on the experiments of [108], [3], [178], [154], it is determined that the transfer of the Haptic feeling through the Internet is feasible. It puzzles out the role of the Internet Service Providers and the QoS that has to be enforced in their networks in order for the user's Quality of Experience to be maximized. The role of the distance and the number of hops between the source and the destination is clarified. It is also obvious that the transfer of Haptic data through WiFi networks is feasible.

12.2 Future work

It has already been scheduled to evaluate the algorithm for real-time transferring through the Internet of HEVC video, audio and haptic streams in real world experiments.

It has also been scheduled to evaluate the presented flow control algorithm in simulations and real world experiments with haptic data transferring. These experiments will have as a primary target to define the proposed values for the factors n, h, j, l, and the initial and the maximum values of κ, m, and qs that the user should set for the sensitivity of the flow control algorithm of [121].

The next step is to propose a transport protocol that is specialized for Haptic virtual environment. This protocol should be efficient and partially reliable. Moreover it should enforce selective and negative acknowledgment, and contain congestion control algorithms. Its main target will be to fulfill the QoS requirements of Table 5-1 for the successful transferring of Haptic data over the Internet.

It will also be important to carry out a comparative study for transport protocol evaluation with more real world experiments, in order to further evaluate real-time transport protocols and estimate the Quality of Experience (QoE) parameters and the Quality of Service (QoS) parameters with a higher degree of accuracy.

12.3 Closing remarks

It is worth pointing out that the research area of haptic protocols is fairly new and open to research and development. The QoS for transferring haptic data is particularly demanding and challenging in order to achieve the maximum QoE. The protocols that are used for haptic transmission are continuously evolving as they are either still in an infant stage, or not specialized for that purpose. Specific protocols for supermedia transferring should be standardized.

The network conditions of the Internet are constantly changing. The metrics such as network delay, jitter and packet loss are not stable. In such a variable environment a flow control algorithm for transferring haptic data is necessary.

It is now clear that the transfer of real-time supermedia data through the Internet is possible. The network conditions of the Internet are continuously improving. The Internet Service Providers should integrate the network conditions that are required for the transfer of haptic data in their QoS.

The new HEVC video encoding is offering great improvements in the supermedia transferring through Internet. The new HEVC technique shows rather promising results as it manages to reduce the bit rate to 50% comparing to its predecessor H.264/AVC. The available Internet bandwidth is no longer an obstacle. The reduced by half bit rate of the video stream provides congestion avoidance and reduced jitter, delay and packet loss.

It is obvious that the synchronization of real-time multimedia and other sensory streams that travel through the Internet is rather a challenging process. The only obstacles in live media streaming is the end to end delay and the jitter of the network. These obstacles are rather obvious when one of the media streams is haptic data, as haptics are very demanding in low delay and jitter.

As far as the Internet of things is concerned, the wireless networks are an integral part of the IoT. The Internet Service Providers should integrate the network conditions that are required for the transfer of supermedia streams in their QoS in order for the IoT to flourish. The 5G mobile network is promising for the future of IoT, as its main goal is to offer bandwidth more than 1 gigabits per second, latency lower than one millisecond, and energy efficiency higher than the 4G.

References

[1] E. Saddik, “The potential of haptics technologies,” *IEEE Instrumentation Measurement Magazine*, vol. 10, pp. 10–17, Feb 2007.

[2] G. Kokkonis, K. Psannis, M. Roumeliotis, and S. Kontogiannis, “A survey of transport protocols for haptic applications,” in *16th Panhellenic Conf. on Informatics (PCI)*, Oct 2012, pp. 192–197.

[3] G. Kokkonis, K. Psannis, M. Roumeliotis, and Y. Ishibashi, “Performance evaluation of transport protocols for real-time supermedia - hevc streams over the internet,” Under Review.

[4] G. Kokkonis, K. Psannis, M. Roumeliotis, S. Kontogiannis, and Y. Ishibashi, “Evaluating transport and application layer protocols for haptic applications,” in *Proc. of. IEEE Intern. Symposium on Haptic Audio-Visual Environments and Games*, 2012, pp. 66–71.

[5] K. Iwata, Y. Ishibashi, N. Fukushima, and S. Sugawara, “Qoe assessment in haptic media, sound, and video transmission: Effect of playout buffering control,” *Computers in Entertainment (CIE)*, vol. 8, no. 2, p. 12, 2010.

[6] R. Marin, P. Sanz, P. Nebot, and R. Wirz, “A multimodal interface to control a robot arm via the web: a case study on remote programming,” *IEEE Trans. Ind. Electron.*, vol. 52, no. 6, pp. 1506–1520, Dec 2005.

[7] P. Huang, Q. Zeng, and Y. Ishibashi, “Qoe assessment of will transmission using haptics: Influence of network delay,” in *Consumer Electronics (GCCE), 2013 IEEE 2nd Global Conference on*, Oct 2013, pp. 456–460.

[8] T. Mizoguchi, T. Nozaki, and K. Ohnishi, “Stiffness transmission of scaling bilateral control system by gyrator element integration,” *IEEE Trans. Ind. Electron.*, vol. 61, no. 2, pp. 1033–1043, Feb 2014.

[9] S. Sakaino, T. Sato, and K. Ohnishi, “Multi-dof micro-macro bilateral controller using oblique coordinate control,” *IEEE Trans. Ind. Informatics*, vol. 7, no. 3, pp. 446–454, Aug 2011.

[10] T. Nozaki, T. Mizoguchi, and K. Ohnishi, “Decoupling strategy for position and force control based on modal space disturbance observer,” *IEEE Trans. Ind. Electron.*, vol. 61, no. 2, pp. 1022–1032, Feb 2014.

[11] S. Hirche and M. Buss, “Human-oriented control for haptic teleoperation,” *Proc. of the IEEE,* vol. 100, no. 3, pp. 623–647, March 2012.

[12] B. Yalcin and K. Ohnishi, “Stable and transparent time-delayed teleoperation by direct acceleration waves,” *IEEE Trans. Ind. Electron.*, vol. 57, no. 9, pp. 3228–3238, Sept 2010.

[13] H. Morimitsu, S. Katsura, and M. Tomizuka, “Design of force compensator with variable gain for bilateral control system under time delay,” in *IEEE Int. Symposium on Ind. Electron. (ISIE),* May 2013, pp. 1–6.

[14] K. Natori and K. Ohnishi, “A design method of communication disturbance observer for time-delay compensation, taking the dynamic property of network disturbance into account,” *IEEE Trans. Ind. Electron.*, vol. 55, no. 5, pp. 2152–2168, May 2008.

[15] O. Linda and M. Manic, “Self-organizing fuzzy haptic teleoperation of mobile robot using sparse sonar data,” *IEEE Trans. Ind. Electron.*, vol. 58, no. 8, pp. 3187–3195, Aug 2011.

[16] K.-B. Sim, K.-S. Byun, and F. Harashima, “Internet-based teleoperation of an intelligent robot with optimal two-layer fuzzy controller,” *IEEE Trans. Ind. Electron.*, vol. 53, no. 4, pp. 1362–1372, June 2006.

[17] I. Peterlik and J. Filipovic, “Distributed construction of configuration spaces for real-time haptic deformation modeling,” *IEEE Trans. Ind. Electron.*, vol. 58, no. 8, pp. 3205–3212, Aug 2011.

[18] D. Wang, K. Tuer, M. Rossi, L. Ni, and J. Shu, “The effect of time delays on tele-haptics,” in *Haptic, Audio and Visual Environments and Their Applications, 2003. HAVE 2003. Proceedings. The 2nd IEEE Internatioal Workshop on*, Sept. 2003, pp. 7 – 12.

[19] K. Hikichi, H. Morino, I. Arimoto, K. Sezaki, and Y. Yasuda, “The evaluation of delay jitter for haptics collaboration over the internet,” in *Global Telecommunications Conference, 2002. GLOBECOM ’02. IEEE*, vol. 2, Nov. 2002, pp. 1492 – 1496.

[20] M. Eid, J. Cha, and A. El Saddik, “Admux: An adaptive multiplexer for haptic–audio–visual data communication,” *IEEE Trans. on Instrumentation and Measurement*, vol. 60, no. 1, pp. 21–31, 2011.

[21] E. Steinbach, S. Hirche, J. Kammerl, I. Vittorias, and R. Chaudhari, "Haptic data compression and communication," *IEEE Signal Processing Magazine*, vol. 28, no. 1, pp. 87–96, 2011.

[22] P. Hinterseer, S. Hirche, S. Chaudhuri, E. Steinbach, and M. Buss, "Perception-based data reduction and transmission of haptic data in telepresence and teleaction systems," *Signal Processing, IEEE Transactions on*, vol. 56, no. 2, pp. 588 – 597, Feb. 2008.

[23] W. keung Fung, N. Xi, W. tai Lo, B. Song, Y. Sun, Y. hui Liu, and I. Elhajj, "Task driven dynamic qos based bandwidth allocation for real-time teleoperation via the internet," in *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, vol. 2, Oct. 2003, pp. 1094 – 1099.

[24] S. Shirmohammadi, N. Woo, and S. Alavi, "Network lag mitigation methods in collaborative distributed simulations," in *Collaborative Technologies and Systems, 2005. Proceedings of the 2005 International Symposium on*, May 2005, pp. 244 –250.

[25] S. Munir and W. Book, "Internet-based teleoperation using wave variables with prediction," *Mechatronics, IEEE/ASME Transactions on*, vol. 7, no. 2, pp. 124–133, Jun. 2002.

[26] F. G. Hamza-Lup and I. A. Stanescu, "The haptic paradigm in education: Challenges and case studies," *The Internet and Higher Education*, vol. 13, no. 1–2, pp. 78 – 81, 2010.

[27] N. Mohammadi and I. Murray, "Developing methodologies for the presentation of graphical educational material in a non-visual form for use by people with vision impairment," in *IEEE Int. Conf. on Teaching, Assessment and Learning for Engineering (TALE)*, Aug 2013, pp. 373–377.

[28] T. Hachisu, G. Cirio, M. Marchal, A. Lécuyer, and H. Kajimoto, "Virtual chromatic percussions simulated by pseudo-haptic and vibrotactile feedback," in *Proceedings of the 8th International Conference on Advances in Computer Entertainment Technology*, ser. ACE '11. New York, NY, USA: ACM, 2011, pp. 20:1–20:5.

[29] A. Guinan, N. Caswell, F. Drews, and W. Provancher, “A video game controller with skin stretch haptic feedback,” in *IEEE Int. Conf. on Consumer Electronics (ICCE)*, Jan 2013, pp. 456–457.

[30] L. Elliott, E. Schmeisser, and E. Redden, “Development of tactile and haptic systems for u.s. infantry navigation and communication,” in *Human Interface and the Management of Information. Interacting with Information*, ser. Lecture Notes in Computer Science, M. Smith and G. Salvendy, Eds. Springer Berlin / Heidelberg, 2011, vol. 6771, pp. 399–407.

[31] M. Hadavand, A. Mirbagheri, H. Salarieh, and F. Farahmand, “Design of a force-reflective master robot for haptic telesurgery applications: Robomaster1,” in *Engineering in Medicine and Biology Society,EMBC, 2011 Annual International Conference of the IEEE*, 30 2011-sept. 3 2011, pp. 7037 –7040.

[32] Y. Nakajima, T. Nozaki, and K. Ohnishi, “Heartbeat synchronization with haptic feedback for telesurgical robot,” *IEEE Trans. Ind. Electron.*, vol. 61, no. 7, pp. 3753–3764, July 2014.

[33] M. Abdur Rahman, A. Alkhaldi, J. Cha, and A. El Saddik, “Adding haptic feature to youtube,” in *Proceedings of the international conference on Multimedia*, ser. MM ’10. ACM, 2010, pp. 1643–1646.

[34] F. Danieau, J. Fleureau, P. Guillotel, N. Mollet, M. Christie, and A. Lecuyer, “Toward haptic cinematography: Enhancing movie experiences with camera-based haptic effects,” *IEEE MultiMedia*, vol. 21, no. 2, pp. 11–21, Apr 2014.

[35] C. H. Park and A. M. Howard, “Real world haptic exploration for telepresence of the visually impaired,” in *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*, ser. HRI ’12. New York, NY, USA: ACM, 2012, pp. 65–72.

[36] J.-M. Normand, M. Servières, and G. Moreau, “A new typology of augmented reality applications,” in *Proceedings of the 3rd Augmented Human International Conference*, ser. AH ’12. New York, NY, USA: ACM, 2012, pp. 18:1–18:8.

[37] S. Hossain, A. Rahman, and A. El Saddik, “Measurements of multimodal approach to haptic interaction in second life interpersonal communication system,” *Instrumentation and Measurement, IEEE Transactions on*, vol. 60, no. 11, pp. 3547 –3558, Nov. 2011.

[38] S. Yajima and S. Katsura, “Multi-dof motion reproduction using motion-copying system with velocity constraint,” *IEEE Trans. Ind. Electron.*, vol. 61, no. 7, pp. 3765–3775, July 2014.

[39] C. J. Sutherland, “An augmented reality haptic training simulator for spinal needle procedures,” 2011.

[40] T. Ando, R. Tsukahara, M. Seki, and M. Fujie, “A haptic interface "force blinker 2" for navigation of the visually impaired,” *IEEE Trans. Ind. Electron.*, vol. 59, no. 11, pp. 4112–4119, Nov 2012.

[41] D. Innala Ahlmark, H. Fredriksson, and K. Hyyppa, “Obstacle avoidance using haptics and a laser rangefinder,” in *IEEE Workshop Advanced Robotics and its Social Impacts (ARSO)*, Nov 2013, pp. 76–81.

[42] K. Suwanratchatamanee, M. Matsumoto, and S. Hashimoto, “Robotic tactile sensor system and applications,” *IEEE Trans. Ind. Electron.*, vol. 57, no. 3, pp. 1074–1087, March 2010.

[43] Y. Oonishi, S. Oh, and Y. Hori, “A new control method for power-assisted wheelchair based on the surface myoelectric signal,” *IEEE Trans. Ind. Electron.*, vol. 57, no. 9, pp. 3191–3196, Sept 2010.

[44] K. Suwanratchatamanee, M. Matsumoto, and S. Hashimoto, “Haptic sensing foot system for humanoid robot and ground recognition with one-leg balance,” *IEEE Trans. Ind. Electron.*, vol. 58, no. 8, pp. 3174–3186, Aug 2011.

[45] R. Wirz, R. Marin, J. M. Claver, M. Ferre, R. Aracil, and J. Fernandez, “End-to-end congestion control protocols for remote programming of robots, using heterogeneous networks: A comparative analysis,” *Robotics and Autonomous Systems*, vol. 56, no. 10, pp. 865 – 874, 2008.

[46] S. Shirmohammadi and N. D. Georganas, “An end-to-end communication architecture for collaborative virtual environments,” *Computer Networks*, vol. 35, no. 3, pp. 351 – 367, 2001.

[47] S. Dodeller and N. D. Georganas, “Transport layer protocols for telehaptics update message,” in *Proc. 22nd Biennial Symposium on Communications, Queen’s Univeristy, Canada*, May 31-June3 2004.

[48] L. Ping, L. Wenjuan, and S. Zengqi, “Transport layer protocol reconfiguration for network-based robot control system,” in *Networking, Sensing and Control, 2005. Proceedings. 2005 IEEE*, Mar. 2005, pp. 1049 – 1053.

[49] Y. Uchimura and T. Yakoh, “Bilateral robot system on the real-time network structure,” *Industrial Electronics, IEEE Transactions on*, vol. 51, no. 5, pp. 940 – 946, Oct. 2004.

[50] R. Wirz, M. Ferre, R. Marin, J. Barrio, J. Claver, and J. Ortego, “Efficient transport protocol for networked haptics applications,” in *Haptics: Perception, Devices and Scenarios*, ser. Lecture Notes in Computer Science, M. Ferre, Ed. Springer Berlin / Heidelberg, 2008, vol. 5024, pp. 3–12.

[51] M. Mauve, V. Hilt, C. Kuhmunch, and W. Effelsberg, “Rtp/i-toward a common application level protocol for distributed interactive media,” *Multimedia, IEEE Transactions on*, vol. 3, no. 1, pp. 152–161, Mar. 2001.

[52] H. Al Osman, M. Eid, R. Iglesias, and A. El Saddik, “Alphan: Application layer protocol for haptic networking,” in *Haptic, Audio and Visual Environments and Games, 2007. HAVE 2007. IEEE International Workshop on*, Oct. 2007, pp. 96 –101.

[53] A. Boukerche, H. Maamar, and A. Hossain, “An efficient hybrid multicast transport protocol for collaborative virtual environment with networked haptic,” *Multimedia Systems*, vol. 13, pp. 283–296, 2008.

[54] S. Floyd, V. Jacobson, S. McCanne, C.-G. Liu, and L. Zhang, “A reliable multicast framework for light-weight sessions and application level framing,” *SIGCOMM Comput. Commun. Rev.*, vol. 25, no. 4, pp. 342–356, Oct. 1995.

[55] S. Paul, K. Sabnani, J.-H. Lin, and S. Bhattacharyya, “Reliable multicast transport protocol (rmtp),” *Selected Areas in Communications, IEEE Journal on*, vol. 15, no. 3, pp. 407 –421, apr 1997.

[56] J. Pullen and V. Laviano, “A selectively reliable transport protocol for distributed interactive simulation,” in *Proceedings of the 13th Workshop on Standards for the Interoperability of Distributed Simulations*, J. Pullen and V. Laviano, Eds., 1995.

[57] R. Wirz, R. Marn, M. Ferre, J. Barrio, J. M. Claver, and J. Ortego, “Bidirectional transport protocol for teleoperated robots,” *ÍEEE Trans. on Industrial Electronics*, vol. 56, no. 9, pp. 3772–3781, 2009.

[58] Z. Cen, M. W. Mutka, D. Zhu, and N. Xi, *NETWORKING 2005. Networking Technologies, Services, and Protocols*, 2005, ch. Supermedia Transport for Teleoperations over Overlay Networks, pp. 1409–1412.

[59] J. P. M. Handley, S. Floyd and J. Widmer, “Tcp friendly rate control (tfrc): Protocol specification,” *RFC*, 2003.

[60] L. Atzori, A. Iera, and G. Morabito, “The internet of things: A survey,” *Computer Networks*, vol. 54, no. 15, pp. 2787 – 2805, 2010.

[61] M. Eraslan, N. D. Georganas, J. R. Gallardo, and D. Makrakis, “A scalable network architecture for distributed virtual environments with dynamic qos over 1pv6,” in *Computers and Communication, 2003. (ISCC 2003). Proceedings. Eighth IEEE International Symposium on*, June 2003, pp. 10–15 vol.1.

[62] E. B. Fgee, J. D. Kenney, W. J. Phillips, W. Robertson, and S. Sivakumar, “Comparison of qos performance between ipv6 qos management model and intserv and diffserv qos models,” in *3rd Annual Communication Networks and Services Research Conference (CNSR’05)*, May 2005, pp. 287–292.

[63] T. O’Reilly and J. Battelle, *Web squared: Web 2.0 five years on*. " O’Reilly Media, Inc.", 2009.

[64] R. I. G. M. INFSO D.4 Networked Enterprise and i. C.-o. w. t. W. G. R. o. t. E. E. Nanosystems, “Internet of things in 2020: Roadmap for the future,” Version 1.1, Tech. Rep., 27 May 2008.

[65] H. Al Osman, J. Cha, and A. El Saddik, “The html5 haptics plugin,” in *IEEE Int. Workshop on Haptic Audio Visual Environments and Games (HAVE)*, Oct 2012, pp. 130–133.

[66] J. Marsh, M. Glencross, S. Pettifer, and R. Hubbold, “A network architecture supporting consistent rich behavior in collaborative interactive applications,” *IEEE Transactions on Visualization & Computer Graphics*, no. 3, pp. 405–416, 2006.

[67] H. Tanaka, K. Ohnishi, H. Nishi, T. Kawai, Y. Morikawa, S. Ozawa, and T. Furukawa, “Implementation of bilateral control system based on acceleration control using fpga for multi-dof haptic endoscopic surgery robot,” *IEEE Trans. Ind. Electron.*, vol. 56, no. 3, pp. 618–627, March 2009.

[68] A. Suzuki, T. Mizoguchi, and K. Ohnishi, “Appropriate controller gain design of acceleration-based bilateral teleoperation under time delay,” in *IEEE Int. Symp. Industrial Electronics (ISIE)*, May 2013, pp. 1–6.

[69] S. Sakaino, T. Sato, and K. Ohnishi, “Precise position/force hybrid control with modal mass decoupling and bilateral communication between different structures,” *IEEE Tran. on Industrial Informatics*, vol. 7, pp. 266–276, May 2011.

[70] J. H. Cho, H. I. Son, D. G. Lee, T. Bhattacharjee, and D. Y. Lee, “Gain-scheduling control of teleoperation systems interacting with soft tissues,” *IEEE Trans. Ind. Electron.*, vol. 60, no. 3, pp. 946–957, March 2013.

[71] H. I. Son, T. Bhattacharjee, and H. Hashimoto, “Effect of impedance-shaping on perception of soft tissues in macro-micro teleoperation,” *IEEE Trans. Ind. Electron.*, vol. 59, no. 8, pp. 3273–3285, Aug 2012.

[72] N. Sakr, J. Zhou, N. Georganas, J. Zhao, and X. Shen, “Prediction-based haptic data reduction and compression in tele-mentoring systems,” in *Instrumentation and Measurement Technology Conference Proceedings, 2008. IMTC 2008. IEEE*, May 2008, pp. 1828–1832.

[73] J. Kammerl, R. Chaudhari, and E. Steinbach, “Combining contact models with perceptual data reduction for efficient haptic data communication in networked ves,” *Instrumentation and Measurement, IEEE Transactions on*, vol. 60, no. 1, pp. 57 –68, jan. 2011.

[74] S. D. Laycock and A. M. Day, “Recent developments and applications of haptic devices,” *Computer Graphics Forum*, vol. 22, no. 2, pp. 117–132, 2003.

[75] A. Roach, *A negative acknowledgement mechanism for signaling compression*, IETF Network Working Group Std., May 2005.

[76] M. Smith and K. Ramakrishnan, “Formal specification and verification of safety and performance of tcp selective acknowledgment,” *Networking, IEEE/ACM Transactions on*, vol. 10, no. 2, pp. 193 –207, Apr 2002.

[77] A. Alturki and M. Alnuem, “Study of the effects of using some qos mechanisms on haptic transmission - using opnet modeler,” in *IEEE Int. Workshop on Haptic Audio Visual Environments and Games (HAVE)*, Oct 2011, pp. 94–101.

[78] Y. Mansour and B. Patt-Shamir, “Jitter control in qos networks,” *IEEE/ACM Trans. Netw.*, vol. 9, no. 4, pp. 492–502, Aug. 2001.

[79] S. Floyd and V. Jacobson, “Link-sharing and resource management models for packet networks,” *Networking, IEEE/ACM Transactions on*, vol. 3, no. 4, pp. 365 –386, aug 1995.

[80] Y.-C. Huang, C.-H. Chu, and E. H.-K. Wu, “A novel congestion control mechanism on tfrc for streaming applications over wired-wireless networks,” in *Proceedings of the 6th ACM workshop on Wireless multimedia networking and computing*, ser. WMuNeP ’11, ACM, 2011, pp. 9–16.

[81] D.-M. Chiu and R. Jain, “Analysis of the increase and decrease algorithms for congestion avoidance in computer networks,” *Computer Networks and ISDN Systems*, vol. 17, no. 1, pp. 1 – 14, 1989.

[82] R. Rejaie, M. Handley, and D. Estrin, “Rap: An end-to-end rate-based congestion control mechanism for realtime streams in the internet,” in *Eighteenth Annual Joint Conf. of the IEEE Computer and Communications Societies. INFOCOM’99.*, vol. 3. IEEE, 1999, pp. 1337–1345.

[83] O. Wongwirat, S. Ohara, and N. Chotikakamthorn, “An adaptive buffer control using moving average smoothing technique for haptic media synchronization,” in *IEEE Int. Conf. on Systems, Man and Cybernetics*, vol. 2, 2005, pp. 1334–1340.

[84] Y. Miyashita, Y. Ishibashi, N. Fukushima, S. Sugawara, and K. E. Psannis, “Qoe assessment of group synchronization in networked chorus with voice and video,” in *TENCON 2011-IEEE Region 10 Conference*, 2011, pp. 192–196.

[85] K. Hosoya, Y. Ishibashi, S. Sugawara, and K. E. Psannis, “Group synchronization control considering difference of conversation roles,” in *IEEE 13th Int. Symposium on Consumer Electronics, ISCE ‘09*, 2009, pp. 948–952.

[86] K. Hosoya, Y. Ishibashi, S. Sugawara, and K. Psannis, “Effects of group synchronization control in networked virtual environments with avatars,” in *Proceedings of the 2008 12th IEEE/ACM Int. Symp. on Distributed Simulation and Real-Time Applications*, 2008, pp. 119–127.

[87] K. Nahrstedt, A. Arefin, R. Rivas, P. Agarwal, Z. Huang, W. Wu, and Z. Yang, “Qos and resource management in distributed interactive multimedia environments,” *Multimedia Tools and Applications*, vol. 51, pp. 99–132, 2011.

[88] I. Psaras and V. Tsaoussidis, “On the properties of an adaptive tcp minimum rto,” *Computer Communications*, vol. 32, no. 5, pp. 888 – 895, 2009.

[89] V. T. Ioannis Psaras, “On the properties of an additive increase rate accelerator,” *Computer Networks*, vol. 53, no. 5, pp. 583 – 595, 2009.

[90] P. Sarolahti, M. Allman, and S. Floyd, “Determining an appropriate sending rate over an underutilized network path,” *Computer Networks*, vol. 51, no. 7, pp. 1815 – 1832, 2007.

[91] M. Rocha, M. Maia, I. Cunha, J. Almeida, and S. Campos, “Scalable media streaming to interactive users,” in *Proceedings of the 13th annual ACM*

international conference on Multimedia, ser. MULTIMEDIA ’05. New York, NY, USA: ACM, 2005, pp. 966–975.

[92] H.-P. Shiang and M. van der Schaar, “Content-aware tcp-friendly congestion control for multimedia transmission,” in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, may 2011, pp. 2356 – 2359.

[93] N. Singhal and R. Sharma, “Survey on tcp friendly congestion control for unicast and multicast traffic,” *International Journal of Computer Applications*, vol. 26, no. 1, pp. 23–30, July 2011, published by Foundation of Computer Science, New York, USA.

[94] N. Suzuki and S. Katsura, “Evaluation of qos in haptic communication based on bilateral control,” in *IEEE Intern. Conference on Mechatronics (ICM)*. IEEE, 2013, pp. 886–891.

[95] E. Isomura, S. Tasaka, and T. Nunome, “A multidimensional qoe monitoring system for audiovisual and haptic interactive ip communications,” in *IEEE Consumer Communications and Networking Conference (CCNC)*. IEEE, 2013, pp. 196–202.

[96] A. Hamam and A. El Saddik, “Toward a mathematical model for quality of experience evaluation of haptic applications,” *Instrumentation and Measurement, IEEE Transactions on*, vol. 62, no. 12, pp. 3315–3322, 2013.

[97] J.-S. Lee, “On designing paired comparison experiments for subjective multimedia quality assessment,” *IEEE Trans. Multimedia*, vol. 16, no. 2, pp. 564–571, Feb 2014.

[98] H. A. Osman, M. Eid, and A. E. Saddik, “Evaluating alphan with multi-user collaboration,” in *Proceedings of the 2008 12th IEEE/ACM International Symposium on Distributed Simulation and Real-Time Applications*, ser. DS-RT ’08. IEEE Computer Society, 2008, pp. 181–186.

[99] J. Zhou, X. Shen, A. E. Saddik, and N. D. Georganas, “Design and implementation of haptic tele-mentoring over the internet,” in *Proceedings of the 11th IEEE International Symposium on Distributed Simulation and Real-Time Applications*, ser. DS-RT ’07. Washington, DC, USA: IEEE Computer Society, 2007, pp. 201–208.

[100] A. Suzuki and K. Ohnishi, “Frequency-domain damping design for time-delayed bilateral teleoperation system based on modal space analysis,” *IEEE Trans. Ind. Electron.*, vol. 60, no. 1, pp. 177–190, Jan 2013.

[101] N. Diolaiti, G. Niemeyer, F. Barbagli, and J. Salisbury, “Stability of haptic rendering: Discretization, quantization, time delay, and coulomb effects,” *Robotics, IEEE Transactions on*, vol. 22, no. 2, pp. 256 –268, April 2006.

[102] P. Huang and Y. Ishibashi, “Qos control and qoe assessment in multi-sensory communications with haptics,” *IEICE Trans. Commun.*, vol. E96-B, no. 2, pp. 392–403, Feb 2013.

[103] C. Demichelis and P. Chimento, *IP packet delay variation metric for IP performance metrics (IPPM)*, IETF RFC 3393 Std., Nov. 2002.

[104] V. Gokhale, O. Dabeer, and S. Chaudhuri, “Hoip: Haptics over internet protocol,” in *IEEE Int. Symp. Haptic Audio Visual Environments and Games (HAVE)*, Oct 2013, pp. 45–50.

[105] F. Brandi and E. Steinbach, “Prediction techniques for haptic communication and their vulnerability to packet losses,” in *IEEE Int. Symp. Haptic Audio Visual Environments and Games (HAVE)*, Oct 2013, pp. 63–68.

[106] T. Asano and Y. Ishibashi, “Adaptive display control of exhibits in a distributed haptic museum,” in *IEEE Int. Symp. Haptic Audio Visual Environments and Games (HAVE)*, Oct 2004, pp. 19–23.

[107] A. Bartl, M. Diaz-Cacho, A. Barreiro, and E. Delgado, “Passivity framework and traffic reduction for the teleoperation of a gantry crane,” in *39th Ann. Conf. IEEE Ind. Electron. Society (IECON)*, Nov 2013, pp. 3675–3680.

[108] G. Kokkonis, K. Psannis, and M. Roumeliotis, “Real time haptic data transferring,” in *IEEE Int. Conf. Wireless Days 2016*, 2016, pp. 1–3.

[109] A. Callado, C. Kamienski, G. Szabo, B. Gero, J. Kelner, S. Fernandes, and D. Sadok, “A survey on internet traffic identification,” *Communications Surveys Tutorials, IEEE*, vol. 11, no. 3, pp. 37–52, 2009.

[110] A. Finamore, M. Mellia, M. Meo, M. Munafo, and D. Rossi, “Experiences of internet traffic monitoring with tstat,” *IEEE Network*, vol. 25, no. 3, pp. 8–14, May 2011.

[111] M. Hasegawa and T. Ikeguchi, “An analysis of the internet traffic by the method of surrogate data,” in *IEEE Int. Symp. Circuits and Systems (ISCAS)*, vol. 3, 2002, pp. III–599–III–602 vol.3.

[112] C. Labovitz, S. Iekel-Johnson, D. McPherson, J. Oberheide, and F. Jahanian, “Internet inter-domain traffic,” *SIGCOMM Comput. Commun. Rev.*, vol. 40, no. 4, pp. 75–86, Aug. 2010.

[113] Y. I. K. E. P. N. Ishii, S. Lee and J. Kim, “Experiment on international connection for haptic media communications,” in *in Record of 2009 Tokai-Section Joint Conf. of the Eight Institutes of Electrical and Related Engineers*, vol. O-434, Sep 2009.

[114] The greek research and technology network—grnet s.a.,. https://www.grnet.gr/en. Accessed 13 May 2015.

[115] R. Garcia and H. Kalva, “Subjective evaluation of hevc and avc/h.264 in mobile environments,” *IEEE Tran. on Consumer Electronics*, vol. 60, pp. 116–123, Feb 2014.

[116] S. McCanne and S. Floyd. Network simulator ns-2. [Online]. Available: http://nsnam.isi.edu/nsnam/index.php

[117] F. Fitzek and M. Reisslein. Video traces for network performance evaluation: Yuv 4:2:0 video sequences. [Online]. Available: http://trace.eas.asu.edu/yuv/-yuv.html

[118] G. Kokkonis, K. E. Psannis, M. Roumeliotis, and Y. Ishibashi, “Efficient algorithm for transferring a real-time hevc stream with haptic data through the internet,” *Journal of Real-Time Image Processing*, pp. 1–13, 2015.

[119] Y. G. K. J. F. S. J. Cao, W.S. Cleveland and M. Weigle, “Stochastic models for generating synthetic http source traffic,” in *IEEE INFOCOM Hong Kong*, March 2004.

[120] M. C. Weigle, P. Adurthi, F. Hernández-Campos, K. Jeffay, and F. D. Smith, “Tmix: A tool for generating realistic tcp application workloads in ns-2,” *SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 3, pp. 65–76, Jul. 2006.

[121] G. Kokkonis, K. E. Psannis, and M. Roumeliotis, *Genetic and Evolutionary Computing: Proceedings of the Ninth International Conference on Genetic and Evolutionary Computing, August 26-28, 2015, Yangon, Myanmar - Volume II*. Springer International Publishing, 2016, ch. Network Adaptive Flow Control Algorithm for Haptic Data Over the Internet–NAFCAH, pp. 93–102.

[122] M. Gerla, M. Y. Sanadidi, R. Wang, A. Zanella, C. Casetti, and S. Mascolo, "Tcp westwood: Congestion window control using bandwidth estimation," in *IEEE Global Telecommunications Conference, GLOBECOM'01.*, vol. 3. IEEE, 2001, pp. 1698–1702.

[123] Y. R. Yang and S. S. Lam, "General aimd congestion control," in *Intern. Conf. on Network Protocols*. IEEE, 2000, pp. 187–198.

[124] J. Widmer, R. Denda, and M. Mauve, "A survey on tcp-friendly congestion control," *IEEE Network*, vol. 15, no. 3, pp. 28–37, 2001.

[125] M. Fujimoto and Y. Ishibashi, "Packetization interval of haptic media in networked virtual environments," in *Proc. of 4th ACM SIGCOMM workshop on Network and system support for games*. ACM, 2005, pp. 1–6.

[126] C. W. Borst, "Predictive coding for efficient host-device communication in a pneumatic force-feedback display," in *Eurohaptics Conf., Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*. IEEE, 2005, pp. 596–599.

[127] S. Lee and J. Kim, "Priority-based haptic event filtering for transmission and error control in networked virtual environments," *Multimedia systems*, vol. 15, no. 6, pp. 355–367, 2009.

[128] Y. Ishibashi, Y. Hashimoto, T. Ikedo, and S. Sugawara, "Adaptive delta-causality control with adaptive dead-reckoning in networked games," in *Proc. of the 6th ACM SIGCOMM workshop on Network and system support for games*. ACM, 2007, pp. 75–80.

[129] P. Hinterseer, E. Steinbach, and S. Chaudhuri, "Perception-based compression of haptic data streams using kalman filters," in *IEEE Intern. Conf. on Acoustics, Speech and Signal Processing*, vol. 5. IEEE, 2006, pp. V–V.

[130] S. Lee and J. Kim, "Dynamic network adaptation scheme employing haptic event priority for collaborative virtual environments," in *Proc. of the First International Conference on Immersive Telecommunications*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2007, p. 12.

[131] A. Aziminejad, M. Tavakoli, R. V. Patel, and M. Moallem, "Transparent time-delayed bilateral teleoperation using wave variables," *IEEE Trans. on Control Systems Technology*, vol. 16, no. 3, pp. 548–555, 2008.

[132] C. Shahabi, A. Ortega, and M. R. Kolahdouzan, "A comparison of different haptic compression techniques," in *IEEE Intern. Conf. on Multimedia and Expo, ICME'02*, vol. 1. IEEE, 2002, pp. 657–660.

[133] S. Allin, Y. Matsuoka, and R. Klatzky, "Measuring just noticeable differences for haptic force feedback: implications for rehabilitation," in *10th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*. IEEE, 2002, pp. 299–302.

[134] E. Karadogan, R. L. Williams, J. N. Howell, R. R. Conatser Jr *et al.*, "A stiffness discrimination experiment including analysis of palpation forces and velocities," *Simulation in Healthcare*, vol. 5, no. 5, pp. 279–288, 2010.

[135] A. J. Silva, O. A. D. Ramirez, V. P. Vega, and J. P. O. Oliver, "Phantom omni haptic device: Kinematic and manipulability," in *Electronics, Robotics and Automotive Mechanics Conference*. IEEE, 2009, pp. 193–198.

[136] M. C. Cavusoglu and F. Tendick, "Multirate simulation for high fidelity haptic interaction with deformable objects in virtual environments," in *IEEE Int. Conf. on Robotics and Automation, ICRA'00*, vol. 3. IEEE, 2000, pp. 2458–2465.

[137] J. Ohm, G. Sullivan, H. Schwarz, T. K. Tan, and T. Wiegand, "Comparison of the coding efficiency of video coding standards - including high efficiency video coding (hevc)," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 22, pp. 1669–1684, Dec 2012.

[138] P. Hanhart, M. Rerabek, F. De Simone, and T. Ebrahimi, "Subjective quality evaluation of the upcoming hevc video compression standard," *Proc. SPIE*, vol. 8499, pp. 84990V–84990V–13, Oct 2012.

[139] J. Nightingale, Q. Wang, and C. Grecos, "Benchmarking real-time hevc streaming," *Proc. SPIE*, vol. 8437, pp. 84370D–84370D–14, Jun 2012.

[140] A. Panayides, Z. Antoniou, M. Pattichis, and C. Pattichis, "The use of h.264/avc and the emerging high efficiency video coding (hevc) standard for developing wireless ultrasound video telemedicine systems," in *Proc. Forty Sixth Asilomar Conf. on Signals, Systems and Computers (ASILOMAR)*, Nov 2012, pp. 337–341.

[141] F. Bossen, B. Bross, K. Suhring, and D. Flynn, "Hevc complexity and implementation analysis," *IEEE Tran. Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1685–1696, Dec 2012.

[142] I.-K. Kim, K. McCann, K. Sugimoto, B. Bross, and H. Woo-Jin, "Hm9: High efficiency video coding (hevc) test model 9 encoder description," in *9th JCT-VC Meeting, Switzerland*, 2012, July, pp. 10–11.

[143] G. Sullivan, J. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (hevc) standard," *IEEE Tran. on Circuits and Systems for Video Technology*, vol. 22, pp. 1649–1668, Dec 2012.

[144] C. C. Chi, M. Alvarez-Mesa, B. Juurlink, G. Clare, F. Henry, S. Pateux, and T. Schierl, "Parallel scalability and efficiency of hevc parallelization approaches," *IEEE Tran. Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1827–1838, Dec 2012.

[145] M. Alvarez-Mesa, C. Chi, B. Juurlink, V. George, and T. Schierl, "Parallel video decoding in the emerging hevc standard," in *IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, Mar 2012, pp. 1545–1548.

[146] Q. Zeng, Y. Ishibashi, N. Fukushima, S. Sugawara, and K. Psannis, "Influences of inter-stream synchronization errors among haptic media, sound, and video on quality of experience in networked ensemble," in *IEEE 2nd Global Conf. on Consumer Electronics (GCCE)*, Oct 2013, pp. 466–470.

[147] T. Schierl, M. Hannuksela, Y.-K. Wang, and S. Wenger, "System layer integration of high efficiency video coding," *IEEE Tran. on Circuits and Systems for Video Technology*, vol. 22, pp. 1871–1884, Dec 2012.

[148] Y. Ishibashi and S. Tasaka, "A comparative survey of synchronization algorithms for continuous media in network environments," in *Proc. 25th Ann. IEEE Conf. on Local Computer Networks*, Nov 2000, pp. 337–348.

[149] Y. Ishibashi, S. Tasaka, and H. Ogawa, "A comparison of media synchronization quality among reactive control schemes," in *IEEE Proc. 14th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, Apr 2001, pp. 77–84 vol.1.

[150] Y. Ishibashi, S. Tasaka, and A. Tsuji, "Measured performance of a live media synchronization mechanism in an atm network," in *IEEE Int. Conf. on Communications*, vol. 3, Jun 1996, pp. 1348–1354.

[151] Y. Ishibashi, T. Kanbara, and S. Tasaka, "Inter-stream synchronization between haptic media and voice in collaborative virtual environments," in *Proc. of the 12th Annual ACM Int. Conf. on Multimedia*, 2004, pp. 604–611.

[152] Y. Ishibashi and S. Tasaka, "A synchronization mechanism for continuous media in multimedia communications," in *IEEE Proc. 14th Annual Joint Conf. of the IEEE Computer and Communications Societies. Bringing Information to People (INFOCOM)*, Apr 1995, pp. 1010–1019 vol.3.

[153] S. Tasaka, T. Nunome, and Y. Ishibashi, "Live media synchronization quality of a retransmission-based error recovery scheme," in *Communications, 2000. ICC 2000. 2000 IEEE International Conference on*, vol. 3, 2000, pp. 1535–1541.

[154] G. Kokkonis, K. Psannis, M. Roumeliotis, and D. Schonfeld, "Real-time wireless multisensory smart surveillance with 3d-hevc streams for internet-of-things (iot)," *The Journal of Supercomputing*, pp. 1–19, Jun. 2016.

[155] I. D. N. Enterprise, R. I. G. Micro, Nanosystems, and W. G. R. of the ETP EPOSS, "Internet of things in 2020: Roadmap for the future," Version 1.1, Tech. Rep., 27 May 2008.

[156] M. Valera and S. A. Velastin, "Intelligent distributed surveillance systems: a review," in *Vision, Image and Signal Processing, IEE Proceedings-*, vol. 152, no. 2. IET, 2005, pp. 192–204.

[157] H.-I. Kim, S. H. Lee, and Y. M. Ro, "Low-power face detection for smart camera," in *Theory and Applications of Smart Cameras*, 2016, pp. 139–155.

[158] R. T. Collins, A. J. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver, N. Enomoto, O. Hasegawa, P. Burt *et al.*, *A system for video surveillance and monitoring*. Carnegie Mellon University, the Robotics Institute Pittsburg, 2000, vol. 2.

[159] N. T. Siebel and S. Maybank, "The advisor visual surveillance system," in *ECCV 2004 workshop applications of computer vision (ACV)*, vol. 1, 2004.

[160] C.-F. Shu, A. Hampapur, M. Lu, L. Brown, J. Connell, A. Senior, and Y. Tian, "Ibm smart surveillance system (s3): a open and extensible framework for event based surveillance," in *IEEE Conf. on Advanced Video and Signal Based Surveillance, 2005. AVSS 2005.*, 2005, pp. 318–323.

[161] E. Durucan, T. Ebrahimi, M. Eckmann, F. Bartolini, A. Tefas, M. Barni, I. Pitas, A. Lipton, H. Fujiyoshi, T. Kanade *et al.*, "Special issue on video communications, processing, and understanding for third generation surveillance systems."

[162] X. Dong and J. Wen, “Low lighting image enhancement using local maximum color value prior,” *Frontiers of Computer Science*, vol. 10, no. 1, pp. 147–156, 2016.

[163] S. Singh, C. Shekhar, and A. Vohra, “Fpga-based real-time motion detection for automated video surveillance systems,” *Electronics*, vol. 5, no. 1, p. 10, 2016.

[164] D.-J. Guo, Z.-M. Lu, and H. Luo, “Multi-channel adaptive mixture background model for real-time tracking,” *J. Inform. Hiding Multimedia Signal Process.*, vol. 7, no. 1, pp. 216–221, 2016.

[165] A. El Maadi and M. S. Djouadi, “Large-scale surveillance system: detection and tracking of suspicious motion patterns in crowded traffic scenes,” *Automatika–Journal for Control, Measurement, Electronics, Computing and Communications*, vol. 60, no. 1, 2016.

[166] Z. Sheng, S. Yang, Y. Yu, A. Vasilakos, J. Mccann, and K. Leung, “A survey on the ietf protocol suite for the internet of things: Standards, challenges, and opportunities,” *Wireless Communications, IEEE*, vol. 20, no. 6, pp. 91–98, 2013.

[167] J. Mongay Batalla, “Advanced multimedia service provisioning based on efficient interoperability of adaptive streaming protocol and high efficient video coding,” *Journal of Real-Time Image Processing*, pp. 1–12, 2015.

[168] E. Kurniawati, C. Lau, B. Premkumar, J. Absar, and S. George, “New implementation techniques of an efficient mpeg advanced audio coder,” *IEEE Tran. on Consumer Electronics*, vol. 50, no. 2, pp. 655–665, May 2004.

[169] Z. Zhang, “Microsoft kinect sensor and its effect,” *IEEE MultiMedia*, vol. 19, no. 2, pp. 4–10, Feb 2012.

[170] P. K. Nathan Silberman, Derek Hoiem and R. Fergus, “Indoor segmentation and support inference from rgbd images,” in *ECCV*, 2012.

[171] J.-S. Lee, Y.-W. Su, and C.-C. Shen, “A comparative study of wireless protocols: Bluetooth, uwb, zigbee, and wi-fi,” in *Industrial Electronics Society, 2007. IECON 2007. 33rd Annual Conference of the IEEE*, Nov 2007, pp. 46–51.

[172] E. H. Ong, J. Kneckt, O. Alanen, Z. Chang, T. Huovinen, and T. Nihtila, “Ieee 802.11ac: Enhancements for very high throughput wlans,” in *IEEE 22nd Int. Symp. on Personal Indoor and Mobile Radio Communications (PIMRC)*, Sept 2011, pp. 849–853.

[173] J. Huang, F. Qian, A. Gerber, Z. M. Mao, S. Sen, and O. Spatscheck, “A close examination of performance and power characteristics of 4g lte networks,” in *Proc. 10th International Conference on Mobile Systems, Applications, and Services*, 2012, pp. 225–238.

[174] J. Andrews, S. Buzzi, W. Choi, S. Hanly, A. Lozano, A. Soong, and J. Zhang, “What will 5g be?” *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 6, pp. 1065–1082, June 2014.

[175] J. Tourrilhes, “Packet frame grouping: improving ip multimedia performance over csma/ca,” in *IEEE Int. Conf. on Universal Personal Communications, 1998*, vol. 2, Oct 1998, pp. 1345–1349.

[176] L. Guiyun, J. Yao, Y. Liu, H. Chen, and D. Tang, “Channel-aware adaptive quantization method for source localization in wireless sensor networks,” *International Journal of Distributed Sensor Networks*, no. 214081, p. 13, 2015.

[177] J. Baek and C. Kim, “An energy-efficient video transport protocol for personal cloud-based computing,” *Journal of Real-Time Image Processing*, pp. 1–8, 2014.

[178] G. Kokkonis, K. Psannis, M. Roumeliotis, Y. Ishibashi, B.-G. Kim, and A. G. Constantinides, “Transferring wireless high update rate supermedia streams over iot,” *IEEE Pervasive Computing*, pp. 1–13, Under Review.

Appendix A

Vita

2003 Dipl. in Electrical and Computer Engineer, Aristotle University of Thessaloniki, Greece

2006 M. Sc. in Information Systems, University of Macedonia, Greece

2016 Ph. D. in Computer Science, University of Macedonia, Greece

Publications in Journals

[1] **G. Kokkonis**, K. E. Psannis, M. Roumeliotis, D. Schonfeld, “Real Time Wireless Multisensory Smart Surveillance With 3D - HEVC Streams for IOT”, The Journal of SuperComputing, pp. 1-19, June 2016.

[2] **G. Kokkonis**, K.E. Psannis, M. Roumeliotis, Y. Ishibashi, "Efficient algorithm for transferring a real-time HEVC stream with haptic data through the internet", Journal of Real-Time Image Processing, pp. 1-13, 2015.

Book chapters

[1] **G. Kokkonis**, K.E. Psannis, M. Roumeliotis, "Network Adaptive Flow Control Algorithm for Haptic Data Over the Internet–NAFCAH", Book chapter of Genetic and Evolutionary Computing, pp. 93-102, Sept. 2015

Publications in Conferences

[1] **G. Kokkonis**, K.E. Psannis, M. Roumeliotis, "Real Time Haptic Data Transferring", Int. Conf. Wireless Days 2016, Mar. 2016

[2] **G. Kokkonis**, K.E. Psannis, M. Roumeliotis, "Experiments for Haptic Data Transferring", 2nd Student Conf. of the Dept. of Applied Informatics, University of Macedonia (FSTEP 2015), Dec. 2015

[3] **G**. **Kokkonis,** K. Psannis, M. Roumeliotis, S. Kontogiannis, Y. Ishibashi, " Evaluating Transport and Application Layer Protocols for Haptic Applications", 11th IEEE International Symposium on Haptic Audio Visual Environments and Games (HAVE 2012), Oct. 08-09 2012

[4] **G**. **Kokkonis,** K. Psannis, M. Roumeliotis, S. Kontogiannis, "A Survey of Transport Protocols for Haptic Applications," 16th Panhellenic Conference on Informatics with international participation (PCI 2012), Oct. 05-07 2012

Under Review Papers

[1] **G. Kokkonis**, K. E. Psannis, M. Roumeliotis, Y. Ishibashi, K. Byung-Gyu, A. G. Constantinides “Transferring Wireless High Update Rate Supermedia Streams Over IoT”, IEEE Pervasive Computing. Under Review

[2] **G. Kokkonis,** K. E. Psannis, M. Roumeliotis, Y. Ishibashi, “Evaluating QoS Performance of Real-Time Protocols for Haptic Applications Through Internet”, Under Review