INTERNET TELEPHONY TRANSPORT PROTOCOL (ITTP): AN EFFICIENT TRANSPORT PROTOCOL FOR VOIP APPLICATIONS

by

MOSLEH MOHAMMAD ABU-ALHAJ

Thesis submitted in fulfillment of the
requirements for the degree of
Doctor of Philosophy

DECEMBER 2011

ACKNOWLEDGMENT

Working as a Ph.D. student in University Sains Malaysia (USM) was a significant as well as challenging experience to me. Numerous people were helpful directly or indirectly in shaping up my academic career. It was hardly possible for me to succeed in my research work without this precious help. Here is a small acknowledgement to the persons all who helped me.

The favour, above all, before all, and after all, is entirely Allah's, to whom my never-ending thanks and praise are humbly due.

Next, I would like to take this opportunity to convey my sincere thanks to my supervisor Prof. Dr. Sureswaran Ramadass for all the help and valuable guidance provided to me during the preparation of this thesis. I consider myself privileged to have had the opportunity to work under his guidance. I am also grateful to mu co-supervisor Dr. Wan Tat Chee, for his help and comments.

Moreover, I would like to thank those who are always in my heart; my father for his endless and continuous encouragement and constant support, my mother for her continuous prayers and inspiration, and my dearest brothers and sisters for always keeping a smile on my face and motivating me all the time.

Last but not least, I would like to convey my appreciation to all NAv6 center members and all my friends who supported me towards the success in my research. A special thanks to my friends Ahmad Abu-Shareha and Yoga Latha for their valuable help.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATION

VoIP	Voice over IP
SIP	Session Initiation Protocol
IAX	InterAsterisk Exchange
RTP	Real-time Transport Protocol
UDP	User Datagram Protocol
IETF	Internet Engineering Task Force
VBR	Variable Bit Rate
CBR	Constant Bit Rate
ITTP	Internet Telephony Transport Protocol
TLPs	Transport Layer Protocols
ITU	International Telecommunication Union
PSTN	Public Switching Telephone Network
RTMTPs	Real-time Media Transfer Protocols
RTLPs	Reliable Transport Layer Protocols
RUTLPs	Unreliable Transport Layer Protocols
UTLPs	Unreliable Transport Layer Protocols
TCP	Transmission Control Protocol
SCTP	Stream Control Transmission Protocol
RDP	Reliable Data Protocol
PR-SCTP	Partial Reliable SCTP
SST	Structured Stream Transport
UDP	User Datagram Protocol
UDP Lite	Lightweight User Datagram Protocol
DCCP	Datagram Congestion Control Protocol
QoS	Quality of Service
NAT	Network Address Translation
NS2	Network Simulation 2
OTcl	Object Oriented Tool Command Language
FTP	File Transfer Protocol
HO	Packet Header Overhead
BWU	Bandwidth Utilization

PROTOKOL PENGANGKUTAN TELEFONI INTERNET (ITTP): SATU PROTOKOL PENGANGKUTAN YANG BERKESAN UNTUK APLIKASI VOIP

ABSTRAK

Sejak beberapa tahun kebelakangan ini, sektor telekomunikasi telah mula menuju ke arah teknologi suara melalui protokol Internet (VoIP). Teknologi VoIP ini menggunakan infrastruktur Internet dan protokol untuk memindahkan data VoIP di antara pihak-pihak pemanggil. Secara tipikal, protokol lapisan aplikasi RTP dan protokol lapisan pengangkutan UDP terikat antara satu sama lain untuk menangani keperluan aplikasi VoIP. Walau bagaimanapun, protokol RTP/UDP digunakan untuk memindahkan semua jenis data aplikasi multimedia masa sebenar seperti sidang video, streaming multimedia, data suara dan sebagainya. Oleh itu, protokol RTP/UDP menyediakan maklumat dan algoritma yang tidak diperlukan oleh aplikasi VoIP. Maklumat dan algoritma yang tidak diperlukan ini telah menyebabkan kelengahan dan kehilangan paket yang mengakibatkan pengurangan kualiti aplikasi VoIP dan overhed paket besar yang mengakibatkan penggunaan lebar jalur yang tidak efisien.

Dalam tesis ini, kami mereka bentuk protokol pengangkutan, dikenali sebagai Protokol Pengangkutan Telefoni Internet (Internet Telephony Transport Protocol, ITTP), khusus untuk membawa data aplikasi VoIP. Tidak seperti RTP/UDP, protokol ITTP hanya memberi maklumat yang diperlukan sahaja untuk pemindahan data suara. Dengan ini, ITTP dapat menghapuskan kelewatan yang berlebihan dan kehilangan paket yang berpunca daripada RTP/UDP, dan ini seterusnya meningkatkan lagi kualiti aplikasi VoIP. Di samping itu, ITTP juga

mengurangkan overhed paket besar yang berpunca daripada RTP/UDP, yang dapat meningkatkan penggunaan lebar jalur.

Bukti matematik dan ujian pelaksaan telah digunakan untuk menunjukkan prestasi ITTP dan membandingkannya dengan protokol RTP/UDP. Keputusan menunjukkan bahawa ITTP merupakan satu protokol yang berkeupayaan memindahkan aplikasi data VoIP, mengurangkan masalah dari protokol RTP/UDP. Sebagai contoh, mengambil kira 8 kbps 'codec' dengan 20 ms masa pemprosesan data kepada paket dan 20 bait saiz paket, penambahan overhed paket besar berkurangan sebanyak 70 peratus, penggunaan jalur lebar telah meningkat sebanyak 10.1 peratus, penggunaan penimbal meningkat sebanyak 29.4 peratus, kehilangan paket berkurangan sebanyak 14 peratus, dan kelewatan dapat dikurangkan sebanyak 19.4 peratus.

INTERNET TELEPHONY TRANSPORT PROTOCOL (ITTP): AN EFFICIENT TRANSPORT PROTOCOL FOR VOIP APPLICATIONS

ABSTRACT

Over the past few years, the telecommunications sector started moving towards Voice over Internet Protocol (VoIP) technology. VoIP technology employs the internet infrastructure and protocols to transfer VoIP data between call parties. Typically, the RTP applications layer protocol and the UDP transport layer protocol are bound together to address the VoIP applications requirements. However, the RTP/UDP protocols are used to transfer all types of real-time multimedia applications data such as video conferencing, multimedia streaming, voice data, and so on. Therefore, the RTP/UDP protocols are providing information and algorithms that are not needed by the VoIP applications. This unneeded information and algorithms are causing superfluous delay and packet loss, which results in reducing the VoIP applications quality, and big packet overhead, which results in inefficient bandwidth utilization.

In this thesis, we designed a transport protocol, named Internet Telephony Transport Protocol (ITTP), dedicated to carry the VoIP applications data. Unlike the RTP/UDP, the ITTP protocol provides only the information required to transfer the voice data. As such, the ITTP eliminates the superfluous delay and packet loss resulting from RTP/UDP, which improves the VoIP applications quality. In addition, the ITTP reduces the packet header overhead resulting from RTP/UDP, which improves the bandwidth utilization.

Mathematical proof and implementation test have been used to demonstrate the ITTP performance and comparing it with the RTP/UDP protocol. The result

showed that the ITTP is a promising protocol to transport the VoIP applications data, shortening the problems resulting from the RTP/UDP protocols. For example, when taking an 8 kbps codec with a 20 ms packetization and 20 byte packet size, the added packet header overhead has decreased 70%, bandwidth utilization has improved 10.1%, buffer utilization has improved 29.4%, packet loss has reduced 14%, and delay has reduced 19.4%.

CHAPTER ONE

INTRODUCTION

Telecommunication technology is one of the most important areas that witnessed a noticeable development in the current era of global technology revolution. In particular, voice telecommunication technology is changing from conventional telephone systems (landline) to voice over Internet Protocol (IP) networks. This new technology is called Voice over IP (VoIP) (NASCIO 2005, Wang, Lin et al. 2009).

VoIP technology exploded in popularity beyond anyone's expectations, causing most service providers to either migrate or plan on migrating from their conventional telephone system infrastructure to a VoIP infrastructure (Boucadair 2009). Universities, enterprises, businesses, and corporate entities have also invested in the development and utilization of VoIP technology. The main reason for its fast adaptation is that it allows calls to be made anywhere around the world at cheaper rates, and sometimes, even for free, compared with conventional telephone systems (Manjur, Abu-Alhaj et al. 2011).

VoIP technology employs Internet infrastructure and protocols to transfer VoIP calls around the world. The Real-time Transport Protocol (RTP) and User Datagram Protocol (UDP) typically work together to transfer VoIP applications data, as well as other types, including all types of real-time multimedia applications data (Abu-Alhaj, Kolhar et al. 2010). Therefore, the RTP and UDP protocols contain some information and functions not needed by VoIP applications (Perkins 2005; Spencer, Shumard et al. 2010). Thus, the 20 bytes RTP/UDP protocols are added to the packet header overhead (Casner and Jacobson 1999; VIVALDIPROJECT 2006;

Abu-Alhaj, Kolhar et al. 2010). As such, (i) inefficient bandwidth utilization results in consumption of Internet bandwidth; (ii) inefficient buffer utilization increases the occurrence of packet loss; and (iii) increase the processing, queuing and transmission time, which increase the delay (Casner and Jacobson 1999; Sze, Liew et al. 2002; VIVALDIPROJECT 2006; Abu-Alhaj, Kolhar et al. 2010).

Researchers have made significant effort to improve VoIP technology quality and bandwidth utilization. The present work contributes in this effort by designing and proposing a new VoIP transport protocol called the Internet Telephony Transport Protocol (ITTP), which is dedicated to transferring VoIP applications data only. The ITTP protocol performs the same function of the RTP/UDP protocols and adds only a 6 bytes header. Hence, the 6 ITTP protocol decreases the overhead resulting from the 20 bytes RTP/UDP protocols. As a result, (i) the reduction in the bandwidth consumption promotes a better Internet bandwidth utilization; (ii) improves the buffer utilization up to 40% as calculated in chapter 5, which decreases the occurrence of packet loss; and (iii) decreases the processing, queuing and transmission time which decrease the delay.

1.1 Background

VoIP technology utilizes the Internet infrastructure to replace conventional telephone systems. Moreover, it utilizes network protocols to transfer calls between parties. Two main categories of protocols are being used in VoIP technology systems, namely signaling protocols and media transfer protocols(Abbasi, Prasad et al. 2005).

The signaling protocols are used to establish and manage sessions between call endpoints. Two standard signaling protocols are used in VoIP technology, namely, H.323 (specifically, H.225) and the Session Initiation Protocol (SIP) (Doong and Wei 2009). Recently, the InterAsterisk Exchange Protocol (IAX) has been introduced as a new signaling protocol. Unlike SIP and H.323, IAX is still not a standard (Abbasi, Prasad et al. 2005). The signaling protocol typically chooses a media transfer protocol that it supports during the session initiation (Perea 2008).

The media transfer protocols are used to exchange media data once the session between call endpoints has been established (Abbasi, Prasad et al. 2005). RTP is specialized to transfer all types of real-time media data, including VoIP data (Perkins 2005). IAX, specifically the IAX mini-frame, can transfer real-time media data as well, and is highly optimized for VoIP calls (Spencer, Shumard et al. 2010).

However, the media transfer protocols, both RTP and the IAX mini-frame, are unable to transfer media data by themselves, which explains why they work on top of transport layer protocols. RTP and the IAX mini-frame typically work in conjunction with the transport layer UDP to transfer VoIP applications data. RTP/UDP protocols are currently dominating VoIP applications, and VoIP applications commonly use them to transfer VoIP data (VIVALDIPROJECT 2006; Westerlund, Johanson et al. 2010).

1.1.1 VoIP Codecs

A voice codec (compression/decompression) is a device or a computer program used to compress voice data. The codec first converts analogue voice data to digital data. This digital data is then compressed using a compression algorithm, which varies from one codec to another, as each codec uses its own compression algorithm. Finally, the compressed data is converted to small frames (voice packet payload), typically between 10 and 30 bytes; the frame size depends on the codec itself as depicted in Table 1.1. Each single frame forms a voice packet payload. Thus, the voice packet payload is very small, and attaching 40 bytes of RTP/UDP/IP header causes a big packet overhead. Two types of codecs exist: the first is the variable bit rate (VBR) codec, which produces variable-size frames. The second type is the constant bit rate (CBR) codec, which produces fixed-size frames (Sze, Liew et al. 2002; VIVALDIPROJECT 2006; Spencer, Shumard et al. 2010).

Table 1.1: Commonly used codecs

Codec	Frame size (Byte)	Bitrate/kbps
G.723.1 (lr)	20	5.3
G.723.1 (hr)	24	6.3
G.729	10	8
G.729A	10	8
G.726	15	24
G.728	10	16
GSM	32.5	13
SpeeX	VBR	VBR
AMR	VBR	VBR

1.2 Problem Statement

Internet protocols provide information and define a set of rules in transferring data between Internet devices. Several alternative protocols work in the transport layer, each of which worked with specific range types of Internet applications. As mentioned in Section 1.1, the transport layer protocol UDP is encapsulated within the application layer protocol RTP when transferring VoIP applications data. RTP and UDP are also used with different types of applications, including all types of real-time multimedia applications. Therefore, RTP and UDP protocols contain many information and functions which are not needed by VoIP applications. Hence, from the viewpoint of networks developer, the combination of RTP and UDP causes a number of problems in VoIP applications, which are as follows:

First, RTP and UDP do not make use of Internet bandwidth efficiently. The typical VoIP packet payload size is only between 10 and 30 bytes. Therefore, attaching 20 bytes of RTP/UDP (12 bytes RTP and 8 bytes UDP) header to this small payload results in a big header size, known as an overhead. The header overhead, which is the relative ratio between the header size and the payload size, varies from 67% to 200%. This value exhibits the wastage of the Internet bandwidth (Sze, Liew et al. 2002; Abu-Alhaj, Kolhar et al. 2010; Spencer, Shumard et al. 2010).

In addition, the RTP and UDP overheads also degrade voice quality because the extra information and functions cause unnecessary processing time that increases the delay and inefficient buffer utilization that increases the packet loss. These exhibit the degradation of voice quality (Shin and Schulzrinne 2009; James and Keith 2010; Spencer, Shumard et al. 2010), which will be explained in detail in Chapter Three.

Moreover, RTP/UDP burden Internet traffic. As stated in (Ash, Hand et al. 2005), 300 million or more calls per day running on the Internet, consume up to 40 gigabits per second for headers alone. Thus, the number of VoIP packets running on the Internet is sizeable compared to Internet traffic, and therefore, problems resulting from VoIP packets will be reflected on the Internet traffic that shares the same link (Hoshi, Tanigawa et al. 1999; Perkins 2005; Boucadair 2009).

The problems identified with RTP/UDP are primarily a result of a large header overhead. To address this, a number of studies have focused on packet multiplexing techniques, which combine multiple payloads (codec frames) in one header to reduce the header overhead and save bandwidth. A higher number of multiplexed payloads will result in increased bandwidth efficiency. Multiplexed packets can come from a single source or from different sources.

However, packet multiplexing has many constraints. First, for multiplexed packets that come from the same source, combining more frames increases the delay in the frames' construction time. Second, for packets multiplexed from different sources, multiplexing techniques require several streams with similar properties. All streams will have the same quality of service because they are transferred over a single IP layer, which makes prioritizing the important stream over the other streams impossible. Moreover, multiplexing is not applicable to individual calls. Thus, bandwidth wastage still occurs (Sze, Liew et al. 2002; Perkins 2005).

The IAX mini-frame was also introduced to overcome the RTP/UDP large header overhead. Like RTP, the IAX mini-frame works in conjunction with the UDP protocol. Although IAX/UDP reduces the header overhead compared with RTP/UDP, it still causes a substantial header, which is between 40% and 120%. Most of this overhead is unnecessary in VoIP applications. Thus, IAX/UDP causes the same problems as RTP/UDP, but at a lesser degree. In addition, the use of the IAX mini-frame has no chance of spreading in VoIP in its current status because H323 and SIP, which are being widely used in all VoIP applications, are not workable with IAX mini-frame (Spencer and Miller 2004; Abbasi, Prasad et al. 2005; Spencer, Shumard et al. 2010).

As a result, current VoIP transfer protocols are burdening VoIP technology applications by imposing unnecessary delays and packet loss and consuming bandwidth. Therefore, a new transport protocol is required to transfer VoIP applications data.

1.3 Objectives

The primary aim of the present thesis is to design and propose a new transport protocol dedicated to VoIP application data, considering the problem resulting from RTP/UDP protocols. However, the new protocol should fulfill the following design considerations:

a. A protocol provides information covering all key VoIP application functions, with minimal fields;
b. A protocol with minimal packet overheads. This will result in improved bandwidth utilization, improved buffer utilization, reduced delays and minimal packet loss.

1.4 Contribution

The current protocols being used to transfer VoIP technology applications data are causing problems to VoIP technology applications. The present work proposes a new protocol called ITTP, which is designed to address the key functions of VoIP applications and overcome the problems inherent in the current protocols. The proposed ITTP provides the following design goals:

a) **Optimality**: ITTP was designed to be optimal in terms of providing information necessary to VoIP application key functions, such as timeliness and smooth delivery, with minimal fields.
b) **Simplicity**: despite its optimality, ITTP is still simple and highly optimized for use in VoIP application calls. The ITTP provides only the

key information necessary for functionality; thus, a small header size and low header overhead, As such:

- Efficient bandwidth utilization is achieved and the consumed bandwidth is reduced;
- Eliminating extra delays and reducing packet loss resulting from current protocols, which improve the voice quality; and
- Improvement in buffers utilization.

Therefore, the proposed ITTP is a promising core transport protocol for VoIP technology applications, which can reduce the current challenges in VoIP technology applications.

1.5 Thesis Outline

This thesis is organized into six chapters. This chapter (**Chapter1**) presents the background principles of the Voice over IP system (VoIP) along with the research objectives and contribution.

Chapter 2 reviews existing literature and fundamental concepts related to this work and issues surrounding it all reviewed. The reasons for proposing and designing a new protocol for the VoIP systems are discussed.

Chapter 3 covers the methodology discussion on how the proposed ITTP protocol was designed. The objectives of design the ITTP protocol on how it handles the RTP/UDP problems were discussed. Lastly, the integration of the ITTP protocol with VoIP protocol stack was clarified.

Chapter 4 gives the implementation details of the ITTP protocol for VoIP system. In addition it shows the integration of the ITTP protocol in the NS2 architecture.

Chapter 5 covers the analysis and discussion of the ITTP protocol mathematically, and its performance through detailed experiments in NS2 simulation.

Finally, **Chapter 6** covers the conclusion of the thesis, and recommendations for further research.

CHAPTER TWO

BACKGROUND AND RELATED WORKS

VoIP technology has started dominating the telecommunications world in recent years. VoIP exploits the current Internet infrastructure and protocols to transfer the VoIP applications data between the call endpoints. Typically, the Real-time Transport Protocol (RTP) is working in conjunction with some of the Transport Layer Protocols (TLPs) to carry the VoIP applications data. The aim of this chapter is to present the capability of the TLPs protocols to transfer the VoIP applications data and the obstacles faced by these protocols. In addition to that, this chapter will show the problems resulting from the TLPs protocols, and the methods used to solve these problems.

2.1 Telecommunication Revolution

Communication is one of the most important needs of mankind. Humans used different types of communication throughout the centuries. At the end of the 19th century, telephony emerged as the turning point in human communication. Telephony transfers the voice conversation as analog signal running over the circuit switching telephone networks, known as Public Switched Telephone Network (PSTN). The PSTN became more reliable throughout its existence and provided high service quality (NASCIO 2005; Farley 2006).

In the second-half of the 20th century, Internet technology emerged as a global computer network to transfer all kinds of data. The development and expansion of the Internet in the last decades conveyed many new services and technologies in many sectors. Voice over IP (VoIP) is one of such technologies. VoIP technology changed the voice conversation from analog signals carried by

PSTN to digital data carried over the Internet. The VoIP technology started dominating the telecommunication sector and replaced the PSTN technology (Leiner, Cerf et al. 1997; NASCIO 2005).

The tremendous growth of VoIP is driven by its several fundamental benefits. Firstly, one of the benefits enjoyed by the user is the substantial cost reduction while making long-distance calls via the Internet. Secondly, VoIP provides a host of advanced communication features like call forwarding, call waiting, voicemail, caller ID, and three-way calling at no extra cost. As compared to normal regular phone services who charge for any extra feature. In addition, from the network operator's viewpoint, the VoIP used compression techniques to reduce the call data rate. Thus the 64-Kb/s PSTN channel, which dedicated to carry one PSTN call, can be used to carry several VoIP calls, which consumes less than 10 Kb/s per call. Moreover, the PSTN channel occupied over the whole call duration. While in the VoIP, the bandwidth is consumed only when the voice data is transfer (Sze, Liew et al. 2002; NASCIO 2005; Abu-Alhaj, Kolhar et al. 2009).

2.2 VoIP Protocols

There is big number of protocols running over the Internet, each of which works with certain types of applications, depending on the application requirements and the protocol properties. Like any other applications, VoIP applications have their own requirements such as timely delivery and smooth delivery. Thus, there are certain protocols used by the VoIP applications. In general, the VoIP protocols are divided into two categories; namely signaling protocols and media transfer protocol (Abbasi, Prasad et al. 2005).

2.2.1 Signaling Protocols

The signaling protocols are used to establish and manage the session between the call endpoints. There are two standard signaling protocols for VoIP, namely H.323 and Session Initiation Protocol (SIP) (Doong and Wei 2009). H.323 was the first signaling protocol used in VoIP. H.323 was developed by the International Telecommunication Union (ITU) not only as a signaling protocol, but also as a complete standard to cover most of the multimedia (audio, video, and data conferencing) communication requirements (Papageorgiou 2001; Basicevic, Popovic et al. 2008; packetizer 2011). Meanwhile, SIP is another standard defined by the Internet Engineering Task Force (IETF). Gradually, SIP protocol has overtaken the place of H.323 protocol and dominates the VoIP applications world. In contrast to H.323, SIP is only a signaling protocol and not a complete architecture for multimedia communication. SIP's main purpose is to initiate and tear down the call session (Papageorgiou 2001; Basicevic, Popovic et al. 2008; packetizer 2011). Recently, InterAsterisk Exchange Protocol (IAX) has been introduced as a new signaling protocol to compete with the SIP protocol. IAX appears to be like SIP in its design. Unlike SIP and H.323, IAX is not a standard yet (Abbasi, Prasad et al. 2005).

2.2.2 Real-time Media Transfer Protocols (RTMTPs)

The RTMTPs main purpose is to transfer the media data over the Internet (Abbasi, Prasad et al. 2005). The RTP is the first standard protocol, introduced by IETF in 1996, specialized to transfer the real-time media data. RTP used to exchange the real-time media data, such as the audio packets, between the call endpoints. Nevertheless, the RTP protocol does not provide mechanisms to ensure timely delivery, smooth delivery, error concealment and correction, and congestion control …etc, leaving this to the application designer. However, the RTP protocol provides other information, such as the timestamp and the sequence number, which used by the applications to ensure timely delivery, smooth delivery, and in-order packets delivery …etc (Schulzrinne and Casner 2003; Perkins 2005). Figure 2.1 shows the RTP header format.

<table>
<tr><th colspan="8">RTP HEADER</th></tr>
<tr><td>BITS</td><td colspan="6">00 15</td><td>16 31</td></tr>
<tr><td>0</td><td>V</td><td>P</td><td>X</td><td>CC</td><td>M</td><td>PT</td><td>SEQUENCE NUMBER</td></tr>
<tr><td>32</td><td colspan="7">TIMESTAMP</td></tr>
<tr><td>64</td><td colspan="7">SSRC IDENTIFIER</td></tr>
<tr><td>96</td><td colspan="7">CSRC IDENTIFIERS
...</td></tr>
</table>

Figure 2.1: RTP header format

The IAX protocol is another protocol used for real-time media transfer over the Internet. IAX was originally designed by Mark Spencer in 1999 for use with the Asterisk open source PBX. IAX includes both signaling protocol and media transfer protocol, thus, its two protocols in one. However, IAX main purpose is to transfer the VoIP calls. The IAX includes many types of messages, called frame. The IAX mini-frame used to transfer the media data The IAX mini-frame was designed to be simple and reduces both overhead and bandwidth consumption (Spencer, Shumard et al. 2010; Manjur, Abu-Alhaj et al. 2011). Figure 2.2 shows the IAX mini-frame header format.

IAX HEADER				
BITS	00	15	16	31
0	F	SOURCE CALL NUMBER	TIMESTAMP	
32	DATA			

Figure 2.2: IAX mini-frame header format

However, RTMTPs work on top of some TLPs to be able to transfer the VoIP applications data. Unfortunately, the combination of the RTMTPs with the TLPs burdens the VoIP applications, because of the header overhead and the unnecessary features. We will refer to these two categories by, the RTMTPs and the TLPs, as the transfer protocol from now on. In this chapter, we aim to study the ability of the transfer protocol to transfer the VoIP applications data. More importantly, this study will highlight the shortages of the transfer protocols in transferring the VoIP data.

2.2.2.1 Timestamp Usage

The timestamp is a key field in the RTMTP protocols, it is used for the following purposes (Perkins 2005; Spencer, Shumard et al. 2010):

First, timeliness VoIP packet delivery. Internet phenomena such as routing, queuing, and congestion, among others cause the packet transit times to be different. Therefore, the packets could be received before or after appropriate play-out times, and the voice play-out could be overlapped or delayed. However, the timestamp must be used to play-out the VoIP packet at the appropriate time. For example, if a system receives audio with a 20 ms packet duration, the packets are then played-out every 20 ms. Hence, packet overlap or delay will be avoided.

Second, overcoming the variability of the received bit rate. As discussed in Chapter 1 Section 1.1.1, some voice codecs produce variable-sized frames, also known as variable bit-rate (VBR). Therefore, the voice packets are received at different time intervals. Accordingly, the timestamp is used to schedule the voice packet play-out at the appropriate time. The timestamp is used to calculate the voice packet payload (frame) duration, and then to schedule the play-out according to the frame duration.

Third, working in conjunction with the arrival time to calculate packet delay variations in the network, also known as jitters. The jitter calculation is given by Equation 2.1.

$$\boldsymbol{D}_{(\boldsymbol{i},\boldsymbol{j})} = (\boldsymbol{T}_{\mathbf{j}} - \boldsymbol{T}_{\mathbf{i}}) - (\boldsymbol{S}_{\mathbf{j}} - \boldsymbol{S}_{\mathbf{i}}) = (\boldsymbol{T}_{\mathbf{j}} - \boldsymbol{S}_{\mathbf{j}}) - (\boldsymbol{T}_{\mathbf{i}} - \boldsymbol{S}_{\mathbf{i}}) \qquad \mathbf{2.1}$$

Where, $D_{(i,j)}$ is the delay jitter for packets *i* and *j*. S_i and S_j are the packets from the source *S* timestamps for packets *i* and *j*, respectively. T_{i} and T_{j} are the arrival times at the target machine *T* for packets *i* and *j*, respectively.

Fourth, reordering out-of-order packets. VoIP packets commonly reach a receiver side in a wrong order. Therefore, the play-out of packets as they are received could cause voice overlaps and disorders. Hence, VoIP packets must be arranged in the order that they were sent before being played out. Accordingly, the RTMTP protocols use the timestamp value enclosed in VoIP packets to chronologically reorder the packets. For example, if the timestamp values of received packets are 10, 20, 60, 50, 30, 40, and 70 ms, respectively, the RTMTP protocol chronologically reorders the packets according to the timestamp values, as in 10, 20, 30, 40, 50, 60, and 70 ms.

2.3 Transport Layer Protocols (TLPs)

The purpose of the transport layer is to provide transparent transfer of data between end users, within a layered architecture of network components and protocols. There are several protocols used in the transport layer, each of which targets different type of applications. In essence, the transport protocols provide the addressing information, typically port-number, to identify the received applications. However, the transport protocols provide different information and support various features and mechanisms to meet the applications requirement, such as the VoIP applications (Noda, Sakai et al. 2002; Kozierok 2005). In this section we will discuss the TLPs from the perspective of VoIP applications. We will focus on the main features of each protocol; concentrating only on the features that affect the VoIP applications. In addition, we will focus on the shortcomings which hinder the usage of the TLPs to transfer the VoIP packets. For a better understanding, we have classified the TLPs into three groups, the Reliable Transport Layer Protocols (RTLPs), Reliable and Unreliable Transport Layer Protocols (RUTLPs), and Unreliable Transport Layer Protocols (UTLPs).

2.3.1 RTLPs Group

This section discusses about the protocols classified as reliable protocols; it shows the main features of each protocol which is related to the VoIP applications. All the RTLPs protocols will be discussed, whether the protocol has already been used with the VoIP applications or not. The reason for discussing all the protocols is to answer the following questions:

- What are the common built-in problems among the RTLPs group protocols that affect the VoIP applications performance?

- Why not adopt a protocol of the RTLPs group?
- Why the application designers avoid using the RTLPs group to transfer the VoIP applications data?
- Why a new VoIP transport protocol is needed?

2.3.1.1 Transmission Control Protocol (TCP)

TCP is a transport layer protocol which has been published as standard RFC by the Internet Engineering Task Force (IETF) in 1981, Figure 2.3 shows the TCP header format. The TCP protocol is the widest spread protocol used in transport layer and it is considered as a mainstay in the Internet communications. The 20bytes TCP protocol contains many features and mechanisms which make it widely used in networks applications. Firstly, The TCP protocol is a connection-oriented protocol, which means that the TCP protocol must establish a connection session between the network endpoints before starting to transfer the data between them. That gives the TCP protocol the ability to manage the connection session between the endpoints. Secondly, the TCP protocol is a reliable protocol where it guarantees the transfer of each single bit without any loss, damages, or duplication. TCP achieves this by sending acknowledgment from the receiver side to the sender side. The acknowledgement is sent after a specific data size has been transmitted. This is called window size. This feature makes the TCP protocol is highly recommended for applications which require high reliability. Thirdly, the TCP protocol guarantees in-order delivery. Where the packets transfer to the other endpoint through different paths, thus, delivered out-of–order. Hence, TCP reorders the packets before sending them to the application layer. Therefore, the TCP features, which provide consistent, trustworthy and securable service to the end users, make it a desirable protocol (Postel 1981; Goode 2002; Zhang and Schulzrinne 2004).

TCP HEADER											
BITS	00									15	16 31
0	SOURCE PORT										DESTINATION PORT
32	SEQUENCE NUMBER										
64	ACKNOWLEDGEMENT NUMBER										
96	DATA OFFSET	RESERVED	CWR	ECE	URG	ACK	PSH	RST	SYN	FIN	WINDOW SIZE
128	CHECKSUM										URGENT POINTER
160 ...	OPTIONS (IF DATA OFFSET > 5) ...										

Figure 2.3: TCP header format

2.3.1.2 Stream Control Transmission Protocol (SCTP)

Figure 2.4 shows the SCTP header format. The SCTP protocol is another noticeable protocol in the RTLPs group. SCTP was developed by the IETF SIGTRAN working group and was published as RFC 2960 in October 2000. Even though SCTP is a relatively new protocol, especially compared to TCP, its usage is widespread among the networks developers. SCTP has many similar features as TCP and some even better features. Reliability and connection-establishment are the two main joint features between TCP and SCTP, the connection known as association in SCTP.

SCTP provides new and great features compared to TCP and all other transport layer protocols. There are three considerable new features. Firstly, the Multi-homing feature which gives SCTP the ability to maintain different associations between the network endpoints, Secondly, the Multi-streaming feature gives the ability to the association to carry multiple streams. Each stream transmits a different type of data. Lastly is the data transmission. SCTP transfers the data as blocks- each block is called a chunk. There are two types of chunks; the first type is the control chunk which is used to control the session. The second type of chunk is the data chunk which is used to send the actual data, which has its own header as well. SCTP header size is 28bytes, 12bytes common header and 16bytes chunk data header (Strewart, Xie et al. 2000; Andreasson, Blanc et al. 2006; Chukarin and Pershakov 2006; Park, Kim et al. 2007).

SCTP HEADER			
BITS	00 ... 15		16 ... 31
0	SOURCE PORT NUMBER		DESTINATION PORT
32	VERIFICATION TAG		
64	CHECK SUM		
96	CHUNK 1 TYPE	CHUNK 1 FLAGS	CHUNK 1 LENGTH
128	CHUNK 1 DATA		
...	...		
...	CHUNK N TYPE	CHUNK N FLAGS	CHUNK N LENGTH
...	CHUNK N DATA		

Figure 2.4: SCTP header format

2.3.1.3 Reliable Data Protocol (RDP)

Figure 2.5 shows the RDP header format. RDP is the last standard transport protocol reviewed in the RTLPs group. RDP has been published as RFC 908 in 1984. After a few years of various experiments on the RDP, another RFC 1151 was published in 1990 to handle the shortcomings of the first RDP issue. However, there is a big similarity between RDP and TCP, where RDP is connection-oriented, reliable, and provide in-order delivery. On the other hand, RDP possesses no new features over TCP. RDP attempts to provide only the necessary functions which make it simpler compared to TCP. In addition, RDP causes less overhead than TCP, because the RDP header size is only 14 bytes. RDP is designed to provide specific type of services such as host monitoring, control applications as loading/dumping and remote debugging (Velten, Hinden et al. 1984; Hinden and Partridge 1990; javvin 2011).

RDP HEADER								
BITS	01							16
0	SYN	ACK	EAK	RST	NUL	0	VER NO	HEADER LENGTH
16	SOURCE PORT							
32	DESTINATION PORT							
48	DATA LENGTH							
64	SEQUENCE NUMBER							
80	ACKNOWLEDGE NUMBER							
96	CHECKSUM							
112	VARIABLE HEADER AREA I							

Figure 2.5: RDP header format

2.3.1.4 RTLPs Group Discussion

In spite of the numerous features and mechanisms, there are several obstacles which make the RTLPs group unsuitable for VoIP applications. The foremost problem is the reliability feature possessed by the RTLPs group. Where, (i) waiting the acknowledgement to send the next window data causes high delay, which is unsuitable to the VoIP applications since they are delay sensitive (ii) retransmission of the lost or damaged packets are futile since these packets are too old to be reintegrated into the stream by the time they are retransmitted. Another important problem is that most of the RTLPs group features and mechanisms are unneeded by the VoIP applications. Therefore, extra unneeded state and processing time, worthless packet overhead, and unjustified implementation complexity. Finally, the RTLPs group protocols have big header weighing to the VoIP packet payload which typically between 10 bytes to 30 bytes. Thus, considerable packet overhead (Larzon, Degermark et al. 1999; Schulzrinne, Casner et al. 2003; Kohler, Handley et al. 2006;

Spencer, Shumard et al. 2010). Table 2.1 shows the packet overhead ratio, added by the RTLPs group in the transport layer. As a result, regardless of the researches on the RTLPs group protocols to carry the VoIP applications data, these unavoidable obstacles make the network developers avoid using the RTLPs group protocols with the VoIP applications. Therefore, design of a new VoIP transport protocol or at least use of other protocols out of this group is needed.

Table 2.1: Overhead ratio: RTLPs group protocols

Protocol	Header Size	Overhead Raito		
		Payload Size 10	Payload Size 20	Payload Size 30
TCP	20	200%	100%	66.6%
SCTP	28	280%	140%	93.3%
RDP	14	140%	70%	46.6%

2.3.2 RUTLPs Group

In this section, we will discuss about the protocols combining both reliability and unreliability features. After discussing the main features of each protocol, we will show the advantage and disadvantage of this group in relation to the VoIP applications. Like the RTLPs group, all the RUTLPs protocols will be discussed, whether the protocol has already been used with the VoIP applications or not. The reason, we discuss all the protocols, is to answer the same questions answered in the RTLPs group section 2.3.1.

2.3.2.1 Partial Reliable SCTP (PR-SCTP)

PR-SCTP is an extension of the SCTP protocol. PR-SCTP is published by IETF as standard RFC in 2004. Two main elements were added to PR-SCTP over SCTP. Firstly, a new parameter is used in the session initiation to determine whether the other endpoint supports the PR-SCTP or not. Secondly, a new control chunk type is used to provide multi levels of the reliability. Hence, the new feature of PR-SCTP over SCTP is that PR-SCTP provides both reliable and unreliable services. Thus, the applications which require unreliable service can benefit from the great features in SCTP. However, the 28 bytes header size is still substantial packet overheads to the VoIP packets. PR-SCTP header format same as SCTP (Molteni and Villari 2002; Stewart, Ramalho et al. 2004).

2.3.2.2 Structured Stream Transport (SST)

SST is a non-standard protocol designed by Bryan Ford, from Massachusetts Institute of Technology, as an experimental transport protocol in November 2007. There is no update to the first release of the SST protocol even though there is no Internet draft submitted to the IETF to make it as a standard protocol. The SST aims to combine the today's network applications requirement in one protocol. Like TCP, SST is a connection-oriented protocol, but the "'Init packets" which sends to initiate a new stream may also contain application data. Therefore, SST does not require a round-trip handshaking delay before the application can begin sending data on a new stream as TCP does. Like SCTP, SST is able to create multiple streams onto a single end-to-end session. SST is considered as flexible protocol, where it supports both reliable and unreliable delivery packet transportation as desired. Moreover, the SST was designed for deployment at two layers namely transport layer alongside TCP and UDP or at application layer running on top of UDP. Furthermore, it supports the data control such as in-order packet delivery or flow control. On the other hand, the SST header size equals 16 bytes, and if it works on top of UDP as usual, the total size will be 24bytes (Ford 2007; UIAproject 2007; PreliminaryProtocolSpecification 2007).

Figure 2.6 and Figure 2.7 show the reliable and unreliable SST header format in transport layer respectively.

SST RELIABLE PROTOCOL HEADER									
BITS	00		15	16					31
0	CHANNEL		TRANSMIT SEQUENCE NUMBER (TSN)						
32	RSVD	ACKCT	ACKNOWLEDGEMENT SEQUENCE NUMBER (ASN)						
64	LOCAL STREAM IDENTIFIER (LSID)			TYPE = 2	-	P	M	C	WINDOW
96	BYTE SEQUENCE NUMBER (BSN)								

Figure 2.6: Reliable SST header format

SST UNRELIABLE PROTOCOL HEADER								
BITS	00		15	16				31
0	CHANNEL		TRANSMIT SEQUENCE NUMBER (TSN)					
32	RSVD	ACKCT	ACKNOWLEDGEMENT SEQUENCE NUMBER (ASN)					
64	LOCAL STREAM IDENTIFIER (LSID)			TYPE = 4	-	B	E	WINDOW

Figure 2.7: Unreliable SST header format

2.3.2.3 RUTLPs Group Discussion

As we can notice, RUTLPs group has advantageous over the RTLPs group. First, it supports both reliable and unreliable delivery. By using the unreliable delivery, this group can avoid the delay resulting from the RTLPs group, which makes it suitable for the real-time VoIP applications. In spit of that, the RUTLPs group still burdening the VoIP applications same as the RTLPs group. Where, RUTLPs group contains many options which cause extra unneeded state and processing time at the end nodes, worthless packet overhead, and unjustified implementation complexity. In addition, the RUTLPs cause substantial overhead to the VoIP applications packets (Kohler, Handley et al. 2006; Spencer, Shumard et al. 2010). Table 2.2 shows the packet overhead ratio, added by RUTLPs group, against the VoIP packet payload. As a result, regardless of the researches on the RTLPs group protocols to carry the VoIP applications data, these unavoidable obstacles make the network developers avoid using the RUTLPs group protocols with the VoIP applications. Therefore, design of a new VoIP transport protocol or at least use of other protocols out of this group is needed.

Table 2.2: Overhead ratio: RUTLPs group protocols

Protocol	Header Size	Overhead Raito		
		Payload Size 10	Payload Size 20	Payload Size 30
PR-SCTP	28	280%	140%	93.3%
SST	16	160%	80%	53.5%
SST/UDP	24	240%	120%	80%

2.3.3 UTLPs Group

In this section, we will discuss about the protocols classified as unreliable protocols. The UTLPs work together with the RTP protocol to transfer the VoIP data. First, we will discuss each of the UTLPs group protocols separately, focusing on the features related to the VoIP. Then, we will discuss the combination of the UTLPs protocols with the RTP protocol. This section will answer the following questions:

- Why the UTLPs group protocols?
- What are the problems resulting when the UTLPs protocols work together with the RTP protocol?
- Why a new VoIP transport protocol is needed?