TRANSPORT LAYER PROTOCOLS TAXONOMY FROM VOICE OVER IP PERSPECTIVE

Mosleh M. Abu-Alhaj[1], Ahmed MANASRAH, Mahmoud Baklizi[1], Nibras Abdullah[1] and Lingeswari V. Chandra[1]

[1]National Advanced IPv6 Center of Excellence

[1]Universiti Sains Malaysia, 11800,

[1]Penang, Malaysia

[1]{mosleh; ahmad; mbaklizi; abdullahfaqera and lingeswari}@nav6.org

ABSTRACT

VoIP technology is becoming increasingly important as they gain market share in PSTN technology. In the short run, it is likely that VoIP technology will dominate the telecommunications market. This evolution towards VoIP technology makes sense for improving the VoIP technology efficiency under the auspices of the internet. However, the PSTN still provide voice quality better than the VoIP technology voice quality. This would be due to many contributing factors such as delay and packet loss. Apart from internet hardware infrastructure, the VoIP transfer protocols cause the delay and packet loss problems as well. Typically, the Real-time Media Transfer Protocols (RTMTPs), such as the Real-time Transport Protocol (RTP), are working in conjunction with the transport layer protocols (TLPs) to carry the VoIP applications data. However, some of the TLPs protocols are transfer the VoIP applications data by themselves.

The aim of this paper is to present the capability of the TLPs protocols to transfer the VoIP applications data, and the obstacles face these protocols to do so. For easier comparison and discussion, we have divided TLPs protocols into three groups, the Reliable Transport Layer Protocols (RTLPs), Unreliable Transport Layer Protocols (UTLPs), and Reliable and Unreliable Transport Layer Protocols (RUTLPs). We have studied the three groups from VoIP point of view. In addition, we have showed that the UTLPs group needs to work with the RTMTPs to be able to transfer the VoIP applications data. Finally, we have showed the problems caused by each group to the VoIP applications.

As result, a comparison among the transport layer protocols is presented according to the problems resulting from theses protocols when transferring the VoIP applications data. The comparison can help to choose the suitable protocols to carry the VoIP data. However, after studied all the TLPs protocols, the paper recommend to use the UTLPs group in conjunction with the RTMTPs, in order to transfer the VoIP applications data.

KEYWORDS

Transport Protocol, RTP, signalling protocol, VoIP

1. INTRODUCTION

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 conservation 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 [1] [2]. Yet it wasn't enough. As humans, we will never be satisfied with the existing technology. 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 conservation 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 [2] [3]. 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 whole the call duration. While in the VoIP, the bandwidth is consumed only when transfer the voice data [2] [4] [5].

2. VoIP PROTOCOLS

There is big number of the 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. Thus, there are certain protocols used by the VoIP applications. In general, the VoIP protocols are divided into two categories [6]. The first category is the signalling protocols. Typically, the signalling protocol is used to establish and manage the session between the call endpoints. There are two standard signalling protocols for VoIP, namely H.323 and Session Initiation Protocol (SIP). H.323 was the first signalling protocol used in VoIP [7]. H.323 was developed by the International Telecommunication Union (ITU) not only as a signalling protocol, but also as a complete standard to cover most of the multimedia (audio, video, and data conferencing) communication requirements [7] [8]. 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 signalling protocol and not a complete architecture for multimedia communication. SIP's main purpose is to initiate and tear down the call session [6] [7] [8]. Recently, InterAsterisk Exchange Protocol (IAX) has been introduced as a new signalling protocol to compete with the SIP protocol. IAX appears to be like SIP in its design. Unlike SIP and H.323, IAX is still not a standard. IAX will be explained in more details later in this paper [6].

The second category is the Real-time Media Transfer Protocols (RTMTPs) [6]. The RTMTPs main purpose is to transfer the media data over the internet. The Real-time Transport Protocol (RTP) is the only 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 depend on the applications needs. However, RTP 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… [9] [10]. Figure1 shows the 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 for use with the Asterisk open source PBX. IAX includes both signalling protocol and media transfer protocol, thus, its two protocols in one. However, IAX main purpose is to transfer the point-to-point VoIP calls, with the ability to handle most types of the media streams. 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. For multiple calls, IAX reduces the overhead of each channel by combining data from several channels into one packet, thus reducing not only the number of

headers, but the number of packets as well. Figure2 shows the IAX mini-frame header format [11] [12] [13].

RTP HEADER								
BITS	00						15	16 31
0	V	P	X	CC	M	PT		SEQUENCE NUMBER
32	TIMESTAMP							
64	SSRC IDENTIFIER							
96	CSRC IDENTIFIERS ...							

Figure1. RTP header format

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

Figure2. IAX mini-frame header format

However, RTMTPs work atop of Transport Layer Protocols (TLPs) to be able to transfer the VoIP applications data. Unfortunately, the combination of the RTMTPs with the TLPs burdens the VoIP applications. We will refer to these two categories, the RTMTPs and the TLPs, as the transfer protocol from now on. In this paper, we aim to study the ability of the transfer protocol to transfer the VoIP applications data. More importantly, this study will highlight the handicaps and the shortages of the transfer protocols in terms of transferring the VoIP data.

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 target 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 different features and mechanisms to meet the applications requirements, such as the VoIP applications [14] [15]. In this section we will discuss the transport layer protocols from the perspective of VoIP. We will focus on the main features of each protocol; with concentrating only the features affect the VoIP applications. In addition, we will focus on the handicaps which hinder the usage of the TLPs to transfer the VoIP packets. For better understanding, we have classified the TLPs into three groups, the Reliable Transport Layer Protocols (RTLPs), Unreliable Transport Layer Protocols (UTLPs), and Reliable and Unreliable Transport Layer Protocols (RUTLPs).

3.1. RTLPs Group

In this section, we will discuss about the protocols classified as reliable protocols. After discussing the main features of each protocol, we will show the reasons behind avoid using this group by the VoIP applications

3.1.1. Transmission Control Protocol (TCP).

Figure 3 shows the TCP header format. TCP is a transport layer protocol which has been published as standard RFC by the Internet Engineering Task Force (IETF) in 1981. 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, TCP is a connection-oriented protocol which means that TCP must establish a session between the network endpoints before starting to transfer the data between them. That gives the TCP the ability to manage the session between the endpoints. TCP uses the well-known three way handshake mechanism to establish the session. Secondly, TCP is a reliable protocol where it guarantees the transfer of each single bit safely without any lost, damages, and 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. The window size is determined by the TCP protocol to avoid the buffer overflow. This feature makes the TCP highly recommended for applications which require high reliability. In addition, 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 [16] [17] [18].

TCP HEADER											
BITS	00							15	16		31
0	SOURCE PORT								DESTINATION PORT		
32	SEQUENCE NUMBER										
64	ACKNOWLEDGEMENT NUMBER										
96	DATA OFFSET	RESERVED	C W R	E C E	U R G	A C K	P S H	R S T	S Y N	F I N	WINDOW SIZE
128	CHECKSUM								URGENT POINTER		
160 ...	OPTIONS (IF DATA OFFSET > 5) ...										

Figure 3. TCP header format.

3.1.2. Stream-Control-Transmission-Protocol (SCTP).

Figure 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 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. However, SCTP uses four-way handshake to set-up connection between the nodes, 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, primary

association and other secondary associations. Thus it transfers the data in the secondary associations in case of any interruption in the primary association. Secondly, the Multi-streaming feature gives the ability to the association to carry multiple streams, where the stream is identified by a unique ID, concurrently. 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. SCTP has many types of control chunk; each type has its own purpose and has its own header. 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 allows transmitting the control and data chunks in the same packet. SCTP header size is 28bytes, 12bytes common header and 16bytes chunk data header. [19] [20] [21] [22].

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 4. SCTP header format.

3.1.3. Reliable Data Protocol (RDP).

Figure 5 shows the RDP header format. RDP is the last standard transport protocol reviewed in the RTLPs group. RDP has been published as RFC in 1984. After a few years of various experiments on the RDP, another RFC was published in 1990 to handle the shortcomings of the first RDP issue. However, there is a big feature 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, where 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. Thus, RDP has very limited usage [23] [24] [25].

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 5. RDP header format.

3.1.4. Discussion

In spite of the numerous features and mechanisms, there are several obstacles which make the RTLPs group unsuitable to 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 at the end nodes, 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 byte. Thus, considerable packet overhead [13] [26] [27]. Table 1 shows the packet overhead ratio, added by the RTLPs group in the transport layer. As result, these obstacles make the network developers avoid using the RTLPs group protocols with the VoIP applications.

Table1. Overhead Ratio-RTLPs group Protocols

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

3.2. UTLPs Group

In this section, we will discuss about the protocols classified as unreliable protocols. After discussing the main features of each protocol, we will show the drivers behind using the VoIP applications to this group. Moreover, we will discuss the shortages of the UTLPs group, and why they need to work in conjunction with the RTMTPs namely RTP protocol and the IAX mini-frame. Finally, we will spotlight the problems resulting from combining the RTLPs group with the RTMTPs.

3.2.1. User Datagram Protocol (UDP)

Figure 6 shows the UDP header format. Like TCP, UDP is also considered as mainstay in the internet communications. UDP is the first standard in the transport protocols; it is published by IETF as standard RFC in 1980. UDP totally differ from TCP in the features. Firstly, UDP is connectionless protocol, thus UDP does not establish any connection between the communication endpoints, and it transfers the data without ensuring that the receiver is available or ready to receive the data. Secondly, UDP is unreliable protocol, which means it transfers the data without any guarantee of the packet delivery, thus, excludes the need of any acknowledgement. Hence, it avoids the delay resulted from the reliable protocols which makes it suitable for real-time applications such as VoIP applications. In addition, UDP is a very simple protocol and does not provide any mechanisms except checksum. Moreover, the UDP is 8 bytes which add small packet overhead, especially if compared to the 20 bytes TCP. On the other hand, UDP does not provide any control to the transferred data. Where, UDP neither provide any mechanism to reorder out-of-order packets nor contain any information for this purpose. Furthermore, it does not recover damaged or lost packets, or even observe the duplicated packets. UDP main purpose is to provide the necessary information to transfer the

14

data between the network nodes, leaving controlling the transferred data to the layer above. However, the simplicity of UDP makes it preferred by wide categories of applications [16] [18] [28] [29].

UDP HEADER			
BITS	00 15	16 31	
0	SOURCE PORT NUMBER	DESTINATION PORT	
32	LENGTH	CHECKSUM	
64	DATA		

Figure 6. UDP header format.

3.2.2. Lightweight User Datagram Protocol (UDP Lite)

Figure 7 shows the UDP-Lite header format. UDP-Lite protocol is IETF RFC standard since 2004. UDP-Lite is similar as the UDP protocol to a very far extent whereas, UDP-Lite is connectionless protocol, unreliable, and does not provide any control on the data such as in-order delivery and recover damaged or lost packets. In other words, UDP-Lite is UDP except that the UDP Lite makes use of partially checksum mechanism to the packet instead of fully checksum. The partially checksum passes the partially damaged packet to the application instead of dropping it like in the fully checksum. This feature is advantageous for wide applications such as the VoIP applications which perform better when dealing with damaged packets rather than dealing with loss packets [26] [30].

UDP LITE HEADER			
BITS	00 15	16 31	
0	SOURCE PORT NUMBER	DESTINATION PORT	
32	CHECKSUM COVERAGE	CHECKSUM	
64	PAYLOAD		

Figure 7. UDP-Lite header format.

3.2.3. Datagram Congestion Control Protocol (DCCP)

Figure 8 shows the DCCP header format. Recently, DCCP has been introduced as a transport protocol which combines features from both TCP and UDP. DCCP is published by IETF as a standard RFC in March 2006. Like TCP, DCCP is a connection-oriented protocol which establishes and tears down the connection between the network endpoints. In addition, DCCP supports congestion control to avoid the cognitive collapse. However, DCCP provides various levels of the congestion control mechanisms whereby the application chooses the suitable congestion control mechanism depending on its requirements. On the other hand, some DCCP properties are similar to the UDP. Where DCCP does not provide reliable delivery or causes any delay in packet transmission. Thus, a DCCP is a suitable choice for applications which requires timely delivery. An example is VoIP application. In addition, DCCP does not support the packet reordering. However, DCCP size is between 12 bytes and 16 bytes depending on the options required. Thus, DCCP causes larger packet overheads to the VoIP applications compared to the 8 bytes of UDP [31] [32].

DCCP HEADER					
BITS	**00**			**15**	**16** **31**
0	SOURCE PORT				DESTINATION PORT
32	DATA OFFSET		CCVAL	CSCOV	CHECKSUM
64	RES	TYPE	X = 1	RESERVED	SEQUENCE NUMBER (HIGH BITS)
96	SEQUENCE NUMBER (LOW BITS)				

Figure 8. DCCP header format.

2.2.4. Discussion

Apparently, the UTLPs group avoids the delay resulting from the reliability features of the RTLPs group. Nevertheless, the UTLPs group provides neither mechanisms nor any information for timely delivery, smooth delivery, and in-order delivery which are key requirements to the VoIP applications. Therefore, the UTLPs group is unable to transfer the VoIP applications packets by themselves. For that reason, the UTLPs group protocols need to be patched by other protocols, namely the RTMTPs, to be able to transfer the VoIP applications packets. There are two protocols for that purpose, the RTP protocol and IAX protocol, specifically the IAX mini-frame.

First, the RTP Protocol: RTP is the most famous protocol used to patch the UTLPs group protocols in order to be able to transfer the VoIP applications packets. However, the combination of the UTLPs group protocols and the RTP protocol cause crucial problems for VoIP applications. First of all is the inefficient use of bandwidth where the combination of the RTP protocol with any of the UTLPs group protocols cause substantial packet overheads compared to the VoIP packets payload. For example, when transmitting a voice stream compressed to 8 kbit/s, each data packet consists of 20 bytes. The result packet overhead, by adding the 20bytes RTP and UDP, will be 100%. However, the incurred overhead increases bandwidth requirements and decreases throughput. Table 2 shows the packet overhead of the UTLPs group protocols when combined to the RTP protocol. Another main problem is degrading the VoIP applications quality. This is due to two main factors. Firstly, the delay, both RTP protocol and UTLPs group protocols contain big number of mechanisms and features which are used to transport various kinds of packets; most of these mechanisms and features are not needed by the VoIP packets. Hence, imposing extra unneeded state and processing overhead at the end nodes [12] [13] [33] [34] [35] [36]. Secondly, the packet loss, where attaching big header size to small payload will result in consumption and decrease in the utilization of the buffer at midway routers and end nodes as well. Thereby increasing the probability of the packets loss. Finally, the big number of mechanisms and features which are not needed by the VoIP applications cause unjustified implementation complexity [12] [33] [37].

Second, the IAX mini-frame: The IAX mini-frame emerged recently to patch the UTLPs group protocols in order to transfer the VoIP packets. However, the IAX mini-frame works only in conjunction with the UDP protocol so far. Unlike RTP, the IAX mini-frame provides only the necessary information to transfer the VoIP packets with UDP. Accordingly, the IAX mini-frame reduces the complexity and processing overhead resulting from the extra options in RTP. Moreover, there are considerable reductions in the packet overheads where IAX mini-frame is 4bytes while the RTP is 12 bytes, Table 2 shows the packet overhead resulting from the IAX mini-frame and the UDP protocol. In spite of that, IAX mini-frame still depends on a transport layer protocol, UDP, in order to transport the VoIP packets. Thus, IAX mini-frame causes the

16

same problems of RTP with less detriment. Most importantly, the IAX mini-header is specified for IAX applications, and does not work with SIP or H323 which they are dominating the VoIP applications [11] 12] [13].

Table2. Overhead Ratio-UTLPs group Protocols and RTMTPs

Transfer Protocol	Size	Overhead Raito		
		Payload Size 10 Bytes	Payload Size 20 Bytes	Payload Size 30 Bytes
IAX mini-frame/UDP	12	120%	60%	40%
RTP/UDP	20	200%	100%	66.6%
RTP/UDP Lite	20	200%	100%	66.6%
RTP/DCCP	24-28	240%-280%	120%-140%	80%-93.3%

3.3. Reliable and Unreliable Transport Layer Protocols (RUTLPs)

In this section, we will discuss about the protocols combined both reliability and unreliability features. After discussing the main features of each protocol, we will show the advantages and disadvantages of this group to the VoIP applications.

3.3.1. 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 used in the session initiation to determine whether the other endpoint supports the PR-SCTP or not. Secondly, a new control chunk type 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 cause substantial packet overheads to the VoIP packets. PR-SCTP header format same as SCTP [38] [39].

3.3.2. 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 as a standard protocol. So, it seems that there is no intention to make it standard so far. 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 host 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, SST supports the data control such as in-order packet delivery or flow control. The SST header size equals 16 bytes, and if it works on top of UDP as usual, the total size will be 24bytes. Figure9 (a) and (b) show the reliable and unreliable SST header format in transport layer respectively [40] [41] [42].

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 9 (a). 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 9 (b). Unreliable SST header format

3.3.3. Discussion

As we can notice, RUTLPs group has advantageous over the RTLPs and UTLPs groups. 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 addition, the RUTLPs group can transfer the VoIP data by itself as its support the basic requirement to do so. In spit of that, the RUTLPs group still burdening the VoIP applications same as the other two groups. Where, RUTLPs group contains many information and options which cause extra unneeded state and processing time, worthless packet overhead, and unjustified implementation complexity [13] [43]. In addition, the RUTLPs cause substantial overhead to the VoIP applications packets. Table 3 shows the packet overhead ratio, added RUTLPs group, against the VoIP packet payload.

Table 3. Overhead Ratio-RUTLPs Protocols

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

4. Findings

VoIP technology is becoming increasingly important as they gain market share in PSTN technology. In the short run, it is likely that VoIP technology will dominate the telecommunications market.   This evolution towards VoIP technology makes sense for improving the VoIP technology efficiency under the auspices of the Internet. However, the PSTN still provide voice quality better than the VoIP technology voice quality. This would be due to many contributing factors such as delay and packet loss in the VoIP technology. Apart from network hardware infrastructure, the VoIP transfer protocols cause the delay and packet loss problems as well.

From VoIP viewpoint, the RTLPs group is not favorable for VoIP applications data transmission due to the large delay. On the other hand, even though the RUTLPs group and UTLPs group in conjunction with RTMTPs group avoid the delay results from the RTLPs group, still these two groups providing unneeded information and features for VoIP applications. In addition, they are considering possibilities far beyond than the current VoIP applications needs. Therefore, worthless overhead, both in terms of the packet header size and in terms of the state and processing overhead. Whereby, degrade the voice quality and inefficient bandwidth utilization.

Accordingly, the current situation of the VoIP transfer protocols is not seen as sustainable state, as VoIP applications clearly have incentives to provide the end users with high voice quality service. Therefore, the VoIP technology needs a new protocol specialized to transport the voice data over the internet. The new protocol should provide the information to address the key VoIP requirements, namely timeliness delivery, in order delivery, and smooth delivery. In addition, it should avoid burdening the VoIP applications by any unneeded information, in order to avoid the aforementioned problems resulting from the existing protocols.

5. Conclusion

In this paper, we have undertaken to study the capability of transfer protocols to transport VoIP packets. We believe that this is the first published research studies and analyze the problems result from transfer the VoIP packets over the transport layer protocols. The research has highlighted that the existing transport layer protocols are obviously causing substantial problems to the VoIP applications. Consequently these hinder the headway of the VoIP applications. Accordingly, the current situation of the VoIP transfer protocols is not seen as sustainable state, as VoIP applications clearly have incentives to provide the end users with high voice quality service. Therefore, the VoIP technology needs a new protocol specialized to transport the voice data over the internet. The new protocol should provide the information to address the key VoIP requirements, namely timeliness delivery, in order delivery, and smooth delivery. In addition, it should avoid burdening the VoIP applications by any unneeded information, in order to avoid the aforementioned problems resulting from the existing protocols.

REFERENCES

[1]   Farley, Tom. "Tom Farley's Telephone History Series. URL: www.privateline.com/TelephoneHistory/History1.htm, published on 2005

[2]   "VoIP and IP Telephony: Planning for Convergence in State Government.", National Association of State Chief Information Officers (NASCIO), May 2005.

[3]   Barry M. Leiner, Vinton G. Cerf, David D. Clark, Robert E. Kahn, Leonard Kleinrock, Daniel C. Lynch, Jon Postel, Lawrence G. Roberts, Stephen S. Wolff , "The past and future history of the Internet ",Communications of the ACM , Volume 40 Issue 2 , 1997

[4]   Abu-Alhaj, M., et al. MuxComp - A New Architecture to Improve VoIP Bandwidth Utilization. in Future Networks, 2009 International Conference on. 2009.

[5]   Sze, H.P., et al., A multiplexing scheme for H.323 voice-over-IP applications. Selected Areas in Communications, IEEE Journal on, 2002. 20(7): p. 1360-1368.

[6]   Abbasi, T.; Prasad, S.; Seddigh, N.; Lambadaris, I., "A comparative study of the SIP and IAX VoIP protocols," Electrical and Computer Engineering, 2005. Canadian Conference on , vol., no., pp.179-183, 1-4 May 2005.

[7]   http://www.packetizer.com/ipmc/h323_vs_sip/, 2010

[8]   Papageorgiou. P, "A Comparison of H.323 vs SIP", University of Maryland at College Par, June 2001.

[9]   Schulzrinne. H., Casner. S , Frederick. R, Jacobson. V, "RTP : A Transport Protocol for Real-Time Applications". IETF RFC 1889. July 2003.

[10]   Addison Wesley," RTP: Audio and Video for the Internet", Edition June 12, 2003

[11]   Spencer, M. and F. Miller, IAX Protocol Description. November l, 2005.

[12]   Boucadair, M., Inter-Asterisk Exchange (IAX): Deployment Scenarios in SIP-Enabled Networks. 1st Edition ed. 2009: John Wiley & Sons Ltd

[13]   Spencer, M., K. Shumard, and B. Capouch, Inter-Asterisk eXchange Version 2, IETF RFC 5456, February 2010

[14]   Noda, Y.; Sakai, T.; Shigeno, H.; Okada, K.; Matsushita, Y.; , "A survey of transport layer protocols suited for real-time data delivery over Diffserv-capable networks," Distributed Computing Systems Workshops, 2002. Proceedings. 22nd International Conference on , vol., no., pp. 305- 310, 2002

[15]   Kozierok, C.M., The tcp/ip guide. Version 3.0 ed. 2005.

[16]   BUR GOODE, "Voice Over Internet Protocol (VoIP)", PROCEEDINGS OF THE IEEE, VOL. 90, NO. 9, SEPTEMBER 2002.

[17]   Marina del Rey, "TRANSMISSION CONTROL PROTOCOL", IETF RFC 793 September 1981.

[18]   Zhang, X., Schulzrinne, H., 2004. Voice over TCP and UDP. Technical Report CUCS-033-04, Computer Science Department, Columbia

[19]   Stewart. R, Morneault. K, Schwarzbauer. H, Rytina. I, Kalla. M, Zhang. L.], Paxson. V," Stream Control Transmission Protocol", IETF RFC 2960. October 2000.

[20]   http://security.maruhn.com/iptables-tutorial/x1535.html.

[21]   Chukarin. A, Pershakov. N. ", Performance Evaluation of the Stream Control Transmission Protocol", IEEE MELECON 2006, May 16-19, Benalmⱶdena (Malaga), Spain.

[22]   ye Lim Parkt, "Myungchul Kimt, and Jeong-Seon Kiml", Evaluation of Stream Control Transmission Protocol as a Transport for VoIP over WLAN", ICACT2007, Feb. 12-14, 2007.

[23]   David Velten, Robert Hinden, Jack Sax ", Reliable Data Protocol", IETF RFC 908. July 1984.

[24]   Partridge. C , Hinden. R," Version 2 of the Reliable Data Protocol (RDP)", IETF RFC 1151. April 1990.

[25]   http://www.javvin.com/protocolRDP.html.

[26]   Larzon. L , Degemark. M, Pink. S, " UDP Lite for Real-Time Multimedia Applications," in Proc. of Sixth IEEE International Workshop on Mobile Multimedia Communications, 1999.

[27]   Schulzrinne. H, Casner. S, Frederick. R, Jacobson. V, "RTP : A Transport Protocol for Real-Time Applications". IETF RFC 1889. July 2003.

[28]   Postel. J, "User Datagram Protocol", IETF RFC 768. August 1980.

[29]   Garg. S, Kappes. M, "An Experimental Study of Throughput for UDP and VoIP Traffic in IEEE 802.11b Networks", IEEE WCNC 2003, Vol. 3, March 2003, pp. 1748-1753.

[30]   Larzon. L.-A, Degemark. M, Pink. S, Jonsson. L.-E, Fairhurst. G, "The Lightweight User Datagram Protocol (UDP-Lite)", IETF RFC 3828, 2004.

[31]   Kohler. E, Handley. M, Floyd. S, " Datagram Congestion Control Protocol (DCCP)", IETF RFC 4340. march 2006.

[32]   Takeuchi. S., Koga. H, Iida. K, Kadobayashi. Y, Yamaguchi. S, "Performance Evaluations of DCCP for Bursty Traffic in Real-time Applications", Proceedings of the The 2005 Symposium on Applications and the Internet (SAINT'05), p.142-149, January 31-February 04, 2005

[33]   Hui Min, C. and H.S. Matthews. Comparative analysis of traditional telephone and voice-over-Internet protocol (VoIP) systems. in Electronics and the Environment, 2004. Conference Record. 2004 IEEE International Symposium on. 2004.

[34]   Subbiah, B., S. Sengodan, and J. Rajahalme. RTP payload multiplexing between IP telephony gateways. in Global Telecommunications Conference, 1999. GLOBECOM '99. 1999.

[35]   Hagsand, O., I. Marsh, and K. Hanson. Sicsophone: A low-delay internet telephony tool. 2003: IEEE.

[36]   The Insufficiency of RTP for TDM Clock Recovery.   2003; Available   from: http://www.dspcsp.com/tdmoip/index.html.

[37]   The Insufficiency of RTP for TDM Clock Recovery.   2003; Available   from: http://www.dspcsp.com/tdmoip/index.html.

[38]   Stewart. R., Ramalho. M., Tuexen. M., Conrad. P, " Stream Control Transmission Protocol (SCTP) Partial Reliability Extension", IETF RFC 3758. May 2004.

[39]   Molteni. M, Villari. M,_"Using SCTP with Partial Reliability for MPEG-4 Multimedia Streaming" BSDCon Europe 2002. October 2002.

[40]   Ford. B," Structured Streams: a New Transport Abstraction", In Proc. of ACM SIGCOMM '07, Kyoto, Japan, Aug. 2007.

[41]   "Structured Stream Transport Preliminary Protocol Specification", PP Specification - 2007 , https://www.pdos.lcs.mit.edu/uia/sst/spec.pdf.

[42]   http://pdos.csail.mit.edu/uia/sst/

[43]   Floyd. S, Handley. M., Kohler. E., "Problem Statement for the Datagram Congestion Control Protocol (DCCP)", IETF RFC 4336, 2006.