

## ITTP: A NEW TRANSPORT PROTOCOL FOR VOIP APPLICATIONS

MOSLEH M. ABU-ALHAJ<sup>1</sup>, MANJUR SK<sup>1</sup>, R. SURESWARAN<sup>1</sup>, TAT-CHEE WAN<sup>1,2</sup>  
IMAD J. MOHAMAD<sup>1</sup> AND AHMED M. MANASRAH<sup>1</sup>

<sup>1</sup>National Advanced IPv6 Centre of Excellence

<sup>2</sup>School of Computer Sciences

Universiti Sains Malaysia

Penang 11800, Malaysia

{ mosleh; manjur; sures; imad; ahmad }@nav6.org; tcwan@cs.usm.my

Received December 2010; revised July 2011

**ABSTRACT.** *Over the past few years, the telecommunications sector has started moving toward the use of Voice over Internet Protocol (VoIP) technology. VoIP technology employs Internet infrastructure and protocols to transfer VoIP data between call parties. Unfortunately, none of the existing Internet transport layer protocols address VoIP application requirements. Typically, the Real-time Transport Protocol (RTP) application layer protocol and the User Datagram Protocol (UDP) transport layer protocol are bound together to address VoIP application requirements. However, a combination of RTP and UDP reduces the quality of VoIP applications and causes inefficient bandwidth utilization. In the present work, a dedicated transport protocol named Internet Telephony Transport Protocol (ITTP) was designed to carry VoIP application data. ITTP is designed to address key VoIP requirements and handle the problems resulting from RTP/UDP. A simple mathematical model was used to evaluate ITTP bandwidth efficiency and compare it with RTP/UDP. ITTP was found to improve bandwidth usage substantially. In addition, ITTP was simulated using Network Simulation 2. The results showed that ITTP had better performance compared with RTP/UDP in terms of packet loss, delay and bandwidth usage.*

**Keywords:** ITTP, VoIP, RTP, VoIP quality, Signaling protocols

**1. Introduction.** In the last decade, network equipment underwent a major performance “revolution”. Network developers took advantage of this change to provide customers with new technologies. Voice over Internet Protocol (VoIP) is one such technology. VoIP utilizes network infrastructure to replace current circuit switching telephone networks called the Public Switched Telephone Network (PSTN) with packet switching telephone networks [1-3]. Furthermore, VoIP technology employs network protocols to transfer calls around the world.

Two main protocol categories are used in VoIP systems [3,4]. The first category comprises the signaling protocols which are used to establish and manage a session between call endpoints [4]. There are two standard signaling protocols for VoIP, namely, H.323 and the Session Initiation Protocol (SIP). H.323 was the first VoIP signaling protocol developed by the International Telecommunication Union (ITU), whereas SIP is a standard developed by the Internet Engineering Task Force (IETF). Gradually, SIP overtook H.323 and dominated the VoIP application world [5,6]. Recently, the InterAsterisk Exchange Protocol (IAX) has been introduced as a new signaling protocol. Unlike SIP and H.323, however, IAX is not yet a standard [4].

The second category comprises the media transfer protocols. Typically, media transfer protocols are used for the exchange of media data once a session is established between

the call endpoints. The Real-time Transport Protocol (RTP) is specialized in transferring all types of real-time media data, including VoIP. IAX, specifically IAX mini-frame, can transfer real-time media data and is optimized highly for VoIP calls. However, media transfer protocols, both RTP and IAX mini-frame, cannot transfer media data by themselves. For this reason, media transfer protocols work atop transport layer protocols. Typically, the transport layer User Datagram Protocol (UDP) works in conjunction with media transfer protocols to transfer VoIP application data [4,7-9].

VoIP technology has started replacing PSTN technology because VoIP provides many advantages for the telecommunications field [10]. The main advantage of VoIP is that it enables calls anywhere around the world at a cheap rate, and sometimes even for free, compared with the conventional PSTN phone system. Second, VoIP enables other functions in addition to voice call, such as video streaming and text messaging, which make users' communication experience more interactive and meaningful. Third, VoIP provides a higher degree of reliability than PSTN. For example, VoIP works around hardware problems automatically, such as out-of-order network hops or damaged network links. Finally, unlike PSTN which is a closed system, VoIP has a free and open architecture, which implies that VoIP extends the opportunity for innovation and creativity to everyone. As a result, the VoIP system continues to undergo rapid and further development [2,7,11].

Nonetheless, network developers perceive that RTP/UDP causes two substantial problems when transporting VoIP data: degradation of voice quality and inefficient bandwidth usage [7,11].

However, VoIP applications must provide phone conversations of better or at least similar quality as the current PSTN technology. In recent years, VoIP developers have made every effort to provide VoIP applications that perform excellently considering the global scale of VoIP technology spread. With respect to these efforts, the present work contributes to the design of a new transport protocol called Internet Telephony Transport Protocol (ITTP), which is dedicated to carrying VoIP application data.

The present paper is organized as follows. Sections 2 and 3 review some related issues and discuss the main drivers behind the design of the ITTP transport protocol. Section 4 discusses the ITTP header and shows the importance of each header field to VoIP applications. Section 5 evaluates ITTP performance using simple mathematical equations and discusses the results. Section 6 implements ITTP and provides simulations showing the protocol performance. Finally, Section 7 presents the conclusion.

**2. Background.** In this section, we review some network issues and VoIP standards that are relevant to our present work, including the RTP and IAX protocols and Network Address Translation (NAT).

**2.1. RTP.** RTP is a standard protocol introduced by the IETF in 1996. Its main purpose is to deliver real-time media, audio or video data over an IP network such as a LAN or WAN. Multimedia conferencing, IP telephony, and video streaming are examples of systems using RTP. However, RTP cannot deliver data by itself. For this reason, RTP works in conjunction with transport layer protocols in order to deliver real-time data. Given this limitation, RTP cannot be considered a transport layer protocol because it works on top of the transport layer protocol. Hence, RTP is best viewed as an application layer protocol. On the other hand, it does not have any mechanism to ensure real-time packet delivery or smooth delivery. Still, RTP provides the required information such as the timestamp and sequence number which are used by real-time application mechanisms to ensure a timely and smooth delivery [12,13].

**2.2. IAX mini-frame.** The IAX protocol has also emerged recently to support media transfer. IAX is designed to be a simple and complete VoIP protocol that can handle both signaling and media transfer. In fact, it is a combined signaling protocol and media transfer protocol. The IAX messages transmitted between the nodes are called frames. There are several types of frame supported by IAX, each of which is used for certain purposes. The IAX mini-frame, which is the focus of the current paper, is used to transfer media data. The design goal for the IAX mini-frame is to minimize bandwidth consumption and to reduce overhead [10,14,15].

**2.3. NAT.** NAT is the process of translating a local IP address into a public IP address. The main purpose for using NAT is to handle the lack of IPv4. Two main types of NAT are used in networks. The first type, known as traditional NAT, only translates the local IP address into a public one. The second type, known as port-based NAT, both translates the local IP address into a public one and changes the port number. A port-based NAT allows multiple connections to share the same public IP address simultaneously by changing the port number. The port-based NAT is often used by all network applications because it saves the IP addresses. Therefore, the new protocol should provide the port number. This issue will be discussed later in the current paper [16,17].

**3. ITTP Design Drivers.** There are many drivers behind the design of ITTP as a new transport layer protocol specialized in carrying VoIP application data.

***First, none of existing transport layer protocols address VoIP application requirements.*** There are many alternative protocols used in the transport layer, each of these protocols carry information to transport specific range types of network applications, depending on the application requirements. Like any other application, VoIP technology applications have specific requirements and needs which must be supported by the transport protocol. The VoIP transport protocol should provide information to facilitate timely and smooth delivery. In addition, the transport protocol should not burden VoIP applications with unnecessary information and mechanisms in order to avoid degrading VoIP application performance [13,18,19].

***Second, RTP and UDP degrade voice quality.*** Typically, the UDP transport layer protocol is patched with the RTP application layer protocol to transfer VoIP data. However, RTP and UDP are used to transfer all types of real-time media data. Therefore, RTP/UDP carries extra information and functions unneeded for VoIP applications, thus resulting in a superfluous processing time, an increase in packet loss, and unjustified complexity, as we will explain in detail later on [13,20].

***Apart from degrading voice quality, RTP/UDP causes an inefficient use of bandwidth in high-cost Internet links.*** The typical VoIP packet payload size is between 10 and 30 bytes. Therefore, attaching 20 bytes of RTP/UDP header, comprising 12 bytes of RTP and 8 bytes of UDP, to this small payload results in a large header size, which is known as overhead. Accordingly, the added overhead which can be calculated using the relative ratio between the header size and the payload size varies from 67% to 200%, thereby wasting Internet bandwidth [7,20,21].

***RTP/UDP burdens Internet links.*** VoIP is gaining popularity, and most telecommunications companies are changing their PSTN infrastructure to VoIP infrastructure or are aiming to do so. Therefore, the number of VoIP application packets running over the Internet will comprise a significant part of the total number of Internet packets. Accordingly, the problems resulting from the use of RTP/UDP will have an effect not only on VoIP application performance but also on the entire link performance. In other words, the VoIP application traffic is sizeable compared with the total Internet traffic. Thus,

the problems resulting from VoIP applications will be reflected on the entire link to some extent [8,10,13].

**The IAX mini-frame causes the same problems as RTP/UDP.** The IAX-mini frame is also used to transfer real-time media data, with high priority given to VoIP data. The 4-byte IAX mini-frame reduces overhead significantly compared with the 12-byte RTP. However, the IAX mini-frame still depends on the UDP transport layer to transfer VoIP data. Thus, IAX/UDP causes the same problems that result from RTP/UDP, although with a lesser effect [10,14,15].

**IAX mini-frame has no chance to spread in the VoIP world.** The two standard signaling protocols, H323 and SIP, dominate the VoIP world. By far, all VoIP applications use H323 and/or SIP as signaling protocols with minimal intention of using the IAX. Unfortunately, IAX mini-frame usage is limited only to IAX applications. Therefore, the IAX mini-frame will not be widely used in the VoIP application world [10,14,15].

**On the other hand, ITTP has a simple design that is highly optimized for VoIP application calls.** ITTP supplies VoIP applications with only the key information that is needed for functionality. All other functionalities are added on the layer above. Hence, the packet overhead resulting from ITTP will be minimal, and the state and processing overhead will be minimal as well. In addition, this simple design gives opportunity to the VoIP developer to add suitable functions and methods for specific application requirements and purposes. On the other hand, ITTP will not be restricted to any signaling protocol, which gives it the opportunity to spread and be adopted by any signaling protocol.

We can conclude that ITTP is needed to transport VoIP application data efficiently. ITTP will be designed as a transport protocol which addresses the necessary VoIP application requirements and handles the problems resulting from the currently used protocols. The following section discusses the ITTP design in detail.

**4. ITTP Protocol Design.** ITTP consists of a 2-byte Source-Port field, 2-byte Destination-Port field, and 2-byte Timestamp field. Figure 1 shows the ITTP header format.

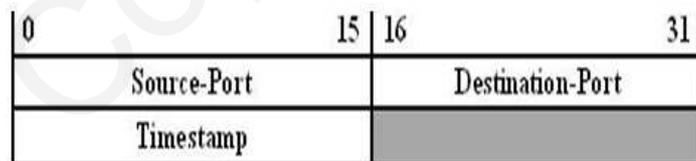


FIGURE 1. ITTP header format

**Source-Port Number.** The source-port number is the transport layer address used to identify the application on the sender endpoint. Usually, the source-port number is used by the receiver endpoint for acknowledgment purposes. However, VoIP applications do not acknowledge the sender endpoint; thus, the source-port number does not serve this purpose. Nevertheless, it is necessary for other purposes, such as the NAT, as explained in the background section.

**Destination-Port Number.** The destination-port number is used to identify the transported application in the receiver endpoint. Each port number within a particular IP device identifies a particular software process.

**Timestamp.** The timestamp is a key field in ITTP. The timestamp represents the number of milliseconds since the first data packet transmission of the call. For each packet, the timestamp increases by a value equal to the packet payload time length. The

timestamp field is 16 bits. Therefore, the maximum value of the timestamp is 65.536 seconds. If the timestamp value reaches 65.536 seconds, then the timestamp will wrap around and again start increasing from the beginning. For example, consider a VoIP application generating one voice packet every 10 milliseconds. The packet timestamp will then be 10, 20, 30, 40, 50, 60, 70, and so on, respectively.

The timestamp is used by VoIP application key functions to ensure real-time and smooth delivery:

***First, the timestamp is used for timely VoIP packet delivery.*** The packet transit time varies because of some issues that take place over the Internet, such as routing, queuing, congestion, and so on. Therefore, the packets may be received before or after the proper play-out time. The voice play-out may overlap or be delayed. However, the timestamp can be used to schedule the play-out of the VoIP application packet at the appropriate time.

***Second, the timestamp is used to overcome the variability of the received bit rate.*** Some voice codecs produce variable-size frames, and thus the variable bit rate. Therefore, the voice packets are received at different time intervals. The timestamp is used to calculate the voice packet payload (frame) duration, and then schedule the play-out based on the frame duration.

***Third, the timestamp is used reorders out-of-order packets.*** A common problem is that VoIP packets reach the receiver endpoint in an incorrect order. Therefore, a play-out of the packets as they are received can cause disorder and voice overlap. Hence, VoIP packets should be ordered in the sequence they were sent before being played out. Accordingly, ITTP uses the timestamp value, which is enclosed within VoIP packets, to arrange the VoIP packets in chronological order.

***Fourth, the timestamp is used to discards duplicate packets.*** In fact, packet duplication is not common in the Internet. However, the timestamp enclosed within the packet is used to check whether the received packet is unique; otherwise, the packet is discarded.

**4.1. Strength of the ITTP design.** As previously mentioned, the current standard uses two protocols, RTP and UDP, to transfer voice media data. UDP works as a data transporter in the transport layer, and RTP works as voice media data carrier in the application layer. ITTP combines the two functions, data transporter and voice media data carrier, into one protocol in the transport layer. In addition, as discussed in the previous section, ITTP provides the key information needed by VoIP application key functionalities. Therefore, ITTP is considered to be two protocols in one compared with the RTP/UDP.

Moreover, ITTP is not restricted to any of the signaling protocols. In other words, it can be adopted by any signaling protocol such as SIP and H323. Thus, ITTP can be used with any VoIP application.

ITTP is also designed to be simple and is highly optimized for VoIP application calls. The 6-byte ITTP performs the same function as the 20-byte RTP/UDP. ITTP provides only the key information (Source Port, Destination Port, and timestamp) needed for VoIP application functionality, leaving all other functionalities to be added on the layer above. The simplicity of ITTP enables it to handle VoIP quality and bandwidth usage efficiency problems resulting from the use of RTP/UDP, as we will discuss in the following section.

As a result, this practical design makes ITTP perform better with VoIP applications compared with other protocols that support many features and provide unnecessary information when transferring a VoIP application call.

**5. ITTP Performance Analysis.** Bandwidth utilization and quality of service are the main concerns in data transformation over the Internet [21,22]. The main goal when designing ITTP is to handle voice quality and bandwidth usage efficiency problems, which result from the use of RTP/UDP. In this section, ITTP performance is analyzed in terms of bandwidth usage efficiency and voice quality. It is also compared with frequently used protocols, RTP and UDP, using a simple mathematical model.

**5.1. Bandwidth usage efficiency.** ITTP bandwidth usage efficiency is evaluated based on the capacity and header overhead. It is also compared with those of RTP and UDP.

**5.1.1. Overhead.** The data over a packet-switched network are fragmented and attached to a header to travel from point to point. The header size varies based on the data type and protocol used. A bigger header size wastes more bandwidth when transferring non-useful data and causes a big packet overhead. As previously mentioned, the 20-byte RTP/UDP is attached to VoIP data in a range of 10 to 30 bytes, thus adding substantial overhead to the VoIP packet. Using Equation (1), the RTP/UDP added overhead ratio can be calculated as follows:

$$\mathbf{Pru\ Overhead\ Ratio} = \frac{\mathbf{Pru}}{\mathbf{f}} * 100\% \quad (1)$$

where  $\mathbf{Pru}$  is the (RTP/UDP) protocol size, and  $\mathbf{f}$  is the frame size in bytes. On the other hand, the 6-byte ITTP performs the same function as RTP/UDP with less overhead and better performance. Using Equation (2), the ITTP overhead ratio can be calculated as follows:

$$\mathbf{Pi\ Overhead\ Ratio} = \frac{\mathbf{Pi}}{\mathbf{f}} * 100\% \quad (2)$$

where  $\mathbf{Pi}$  is the ITTP size in bytes. Using substitutions in Equations (1) and (2), Figure 2 depicts the overhead ratio of RTP/UDP and ITTP with different codec sizes.

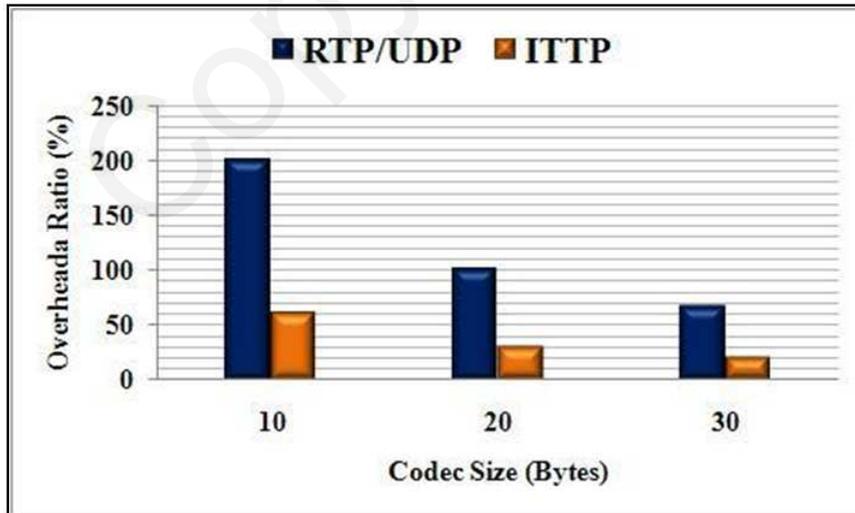


FIGURE 2. Overhead ratio for RTP/UDP and ITTP

As seen in Figure 2, the overhead ratio added by RTP/UDP is 200%, whereas that by ITTP is 60% when the frame size is 10 bytes. When the frame size is 20 bytes, the overhead ratio added by RTP/UDP is 100%, whereas that by ITTP is 30%. When the frame size is 30 bytes, the overhead ratio added by RTP/UDP is 67%, whereas that by ITTP is 20%. As a result, ITTP shows a substantial reduction in the header overhead

with all three frame sizes compared with RTP/UDP. This result is obviously due to the difference in protocol header size, with the RTP/UDP header size being 20 bytes and that of ITTP being only 6 bytes.

5.1.2. *Capacity.* Capacity is the maximum number of concurrent calls that can be run in a specific channel bandwidth. By knowing the per-call bandwidth consumption and the channel bandwidth, the following inequality can be utilized when computing the number of calls in a specific channel:

$$\text{Aggregate } C_{bw} < Ch_{bw} \quad (3)$$

where  $C_{bw}$  is the per-call bandwidth consumption in bps, and  $Ch_{bw}$  is the available channel bandwidth in bps. The per-call bandwidth consumption is calculated using Equation (4):

$$C_{bw} = Pkt_s * PPS = (P_h + F) * \frac{C_{br}}{F} = \frac{\left( (P_h + F) * \frac{C_{br}}{F * 8} \right) * 8}{1000} \quad (4)$$

where  $Pkt_s$  is the packet size in bytes,  $PPS$  is the number of packets per second,  $P_h$  is the packet header size,  $F$  is the frame size, and  $C_{br}$  is the codec bit rate per second. We multiply by 8 to convert from byte to bit, and we divide by 1000 to convert from bit to kilobit. From Equations (3) and (4), the number of concurrent calls running in the same channel bandwidth with ITTP increases compared with that of RTP/UDP. Figure 3 shows the capacity of RTP/UDP and ITTP using an 8000 bps codec bit rate with 10-byte, 20-byte, and 30-byte frames.

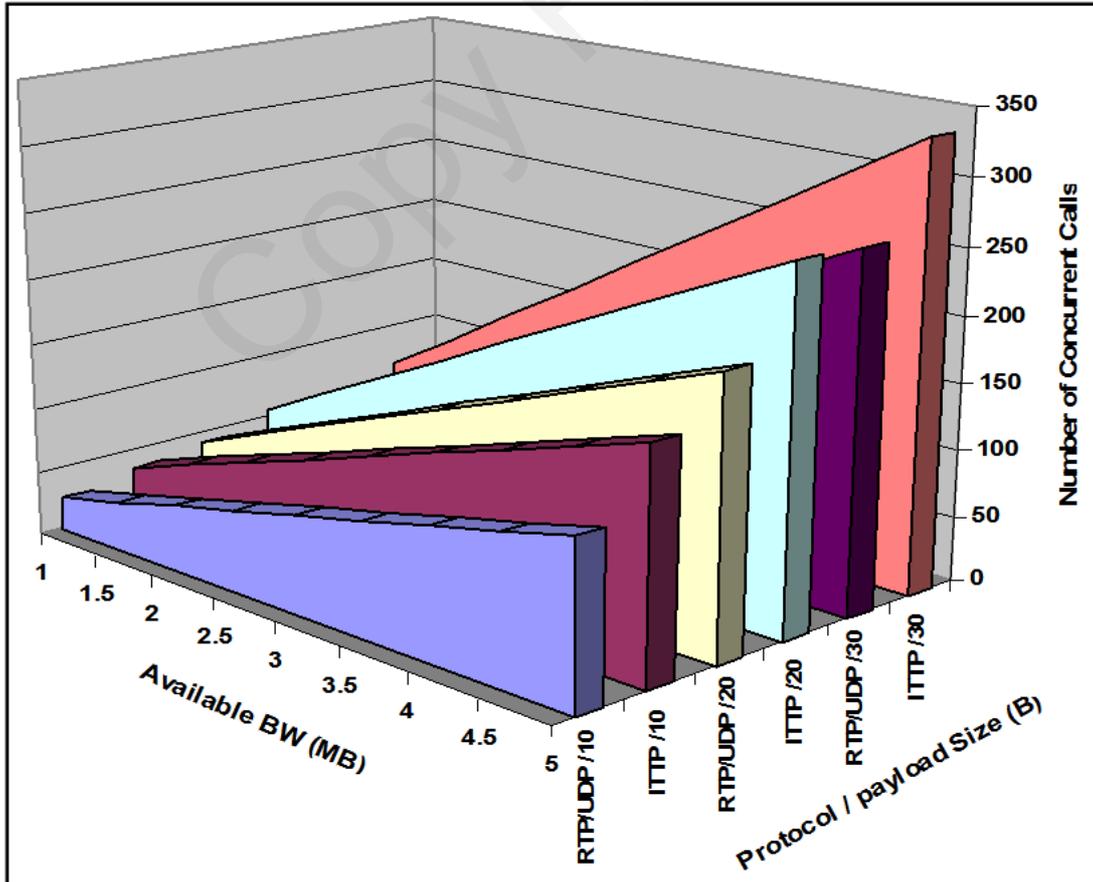


FIGURE 3. Number of concurrent calls

As seen from Figure 3, the number of concurrent calls when using ITTP is greater than that when using RTP/UDP for all three frame sizes. In addition, the difference in the number of concurrent calls increases when the available bandwidth increases. Therefore, the difference in the number of the concurrent calls between ITTP and RTP/UDP is directly proportional to the available bandwidth. For example, the number of concurrent calls when using ITTP at 1 Mb of bandwidth with a 30-byte frame size is 67, whereas that when using RTP/UDP under the same conditions is 53. Therefore, the difference in the number of concurrent calls is 14. On the other hand, the number of concurrent calls when using ITTP at 5 Mb of bandwidth with a 30-byte frame size is 335, whereas that when using RTP/UDP under the same conditions is 268. Therefore, the difference in the number of concurrent calls is 67.

As a result, the overhead and the capacity reflect the bandwidth usage efficiency. ITTP performs the same function as RTP/UDP but with a much better bandwidth usage efficiency.

**5.2. Voice quality.** The transfer of voice traffic over IP is rapidly gaining acceptance. However, end users are concerned about the possible degradation in voice quality when calls are carried over IP networks. There are two main factors affecting voice quality, namely, delay and packet loss [1,23].

**5.2.1. Delay.** RTP/UDP causes extra delay compared with ITTP because of the following reasons:

ITTP reduces the packetization delay resulting from the currently used RTP/UDP. It imposes less message passing between the vertical and horizontal layers compared with RTP/UDP. This lessens the state and processing time at the end nodes. Moreover, unlike ITTP which is specialized in carrying VoIP data, both RTP and UDP headers contain fields unnecessary to VoIP applications. This adds unneeded state and processing overhead at the end nodes. In addition, ITTP reduces the processing delay resulting from RTP/UDP, whereas RTP/UDP causes more delays in the intermediate routers. This is because the time required to process the two protocols, RTP and UDP, which have more fields and larger sizes, is greater than the time required to process one protocol, ITTP, which has fewer fields and a smaller size. Moreover, ITTP reduces the transmission delay resulting from RTP/UDP because the ITTP packet size is smaller than that of RTP/UDP. Finally, ITTP reduces the queuing delay resulting from RTP/UDP because a lesser number of packet-bytes are required to be queued and transferred in each ITTP stream. In addition, the RTP/UDP stream requires more bandwidth than the ITTP stream. Therefore, the number of RTP/UDP packets that queue in the buffer will be more than those for ITTP in case the output bandwidth is less than the input data rate [10,24-27].

All these reasons consequently result in a decrease in the overall delay caused by ITTP compared with RTP/UDP.

**5.2.2. Packet loss.** RTP/UDP causes a big packet overhead, and thus inefficient utilization of the intermediate router buffers, which increases the packet loss. In contrast, ITTP causes a smaller overhead, and thus a more efficient utilization of the intermediate routers and end node buffers. Therefore, ITTP decreases packet loss compared with RTP/UDP. Moreover, ITTP's per-packet processing time is faster than that of RTP/UDP, thus decreasing packet loss as well [1,22,28].

Obviously, ITTP is effective in improving voice quality and bandwidth usage efficiency, especially compared with the currently used protocols, RTP and UDP. Furthermore, ITTP is much easier to implement than RTP/UDP because it is the protocol specialized for VoIP applications. Table 1 summarizes the factors affected by ITTP.

TABLE 1. ITTP performance

Element	Effect
Header overhead	Considerable header overhead reduction
Capacity	Increases the number of concurrent calls running in a specific channel
Bandwidth Usage Efficiency	Improve the bandwidth utilization
Delay	Reduce the delay
Packet loss	Reduce the packet loss
Voice quality	Improve the overall voice quality
Buffer Utilization	Improve the buffer utilization

**6. Implementation Result and Discussion.** The Network Simulation 2 (NS2) tool was utilized to simulate ITTP and to demonstrate its performance. After simulating ITTP, two different network topologies using NS2 were designed to evaluate it. The purpose of this implementation is to evaluate ITTP voice quality and bandwidth usage efficiency compared with those of RTP/UDP.

**6.1. Topology 1: Short-distance calls.** Topology 1 was utilized to evaluate ITTP performance for short-distance calls, where the distance between call ends was three hops. Four different experiments with different parameters were run on Topology 1. In the first experiment, ITTP packet loss was investigated and compared with that of RTP/UDP. In the second experiment, ITTP delay was investigated and compared with that of RTP/UDP. In the third experiment, the number of calls supported by ITTP and that supported by RTP/UDP at various channel bandwidths were compared. In the fourth experiment, the ITTP goodput was calculated and compared with that of RTP/UDP. In Topology 1, the parameters for all streams in each experiment are the same. Therefore, the results constantly increase or decrease, as the results will show.

**6.1.1. Topology 1 setup.** Topology 1 consists of 60 PCs and 5 routers. All LAN links between the PCs and routers have a delay of 2 ms. All WAN links between routers have a delay of 18 ms. Each node uses a DropTail queue, of which the maximum size is 50. A CBR traffic generator is attached to each PC. The data rate upon using RTP/UDP and ITTP is 24 kbps and 18.4 kbps, respectively. The rate varies because of the different header sizes. In the four experiments, the sources started generating traffic at 100 ms and stopped at 1000 ms. Figure 4 depicts Topology 1.

**6.1.2. Experiment 1: Packet loss.** In this experiment, the packet loss of ITTP and RTP/UDP was investigated at various stream numbers between 1 and 8. All links have a bandwidth of 100 kbps. Figure 5 depicts the packet loss ratio of both ITTP and RTP/UDP. That for RTP/UDP started when the number of streams reached 6, whereas that for ITTP started when the number of streams reached 8. This is because the RTP/UDP stream data rate is higher than the ITTP stream data rate. Thus, the RTP/UDP stream consumes more bandwidth than the ITTP stream. Hence, the available bandwidth can carry more ITTP streams than RTP/UDP streams.

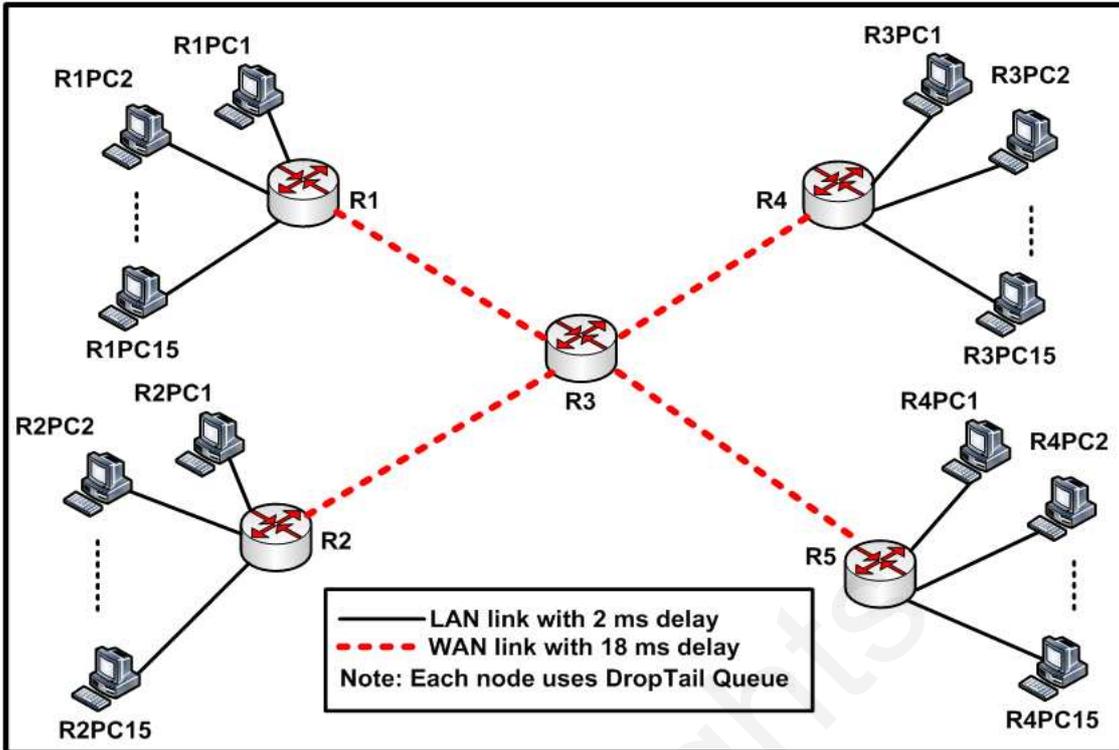


FIGURE 4. Topology 1

Moreover, the packet loss for ITTP is less than that for RTP/UDP under similar conditions because of two main reasons. First, RTP/UDP causes a higher packet overhead than ITTP. Therefore, the RTP/UDP packet size is bigger than the ITTP packet size. As a result, the intermediate router buffer can store more ITTP packets than RTP/UDP packets. Therefore, the packet loss is decreased compared with RTP/UDP. Second, the processing time for ITTP packets is less than the processing time for RTP/UDP packets, thereby decreasing the packet loss as well.

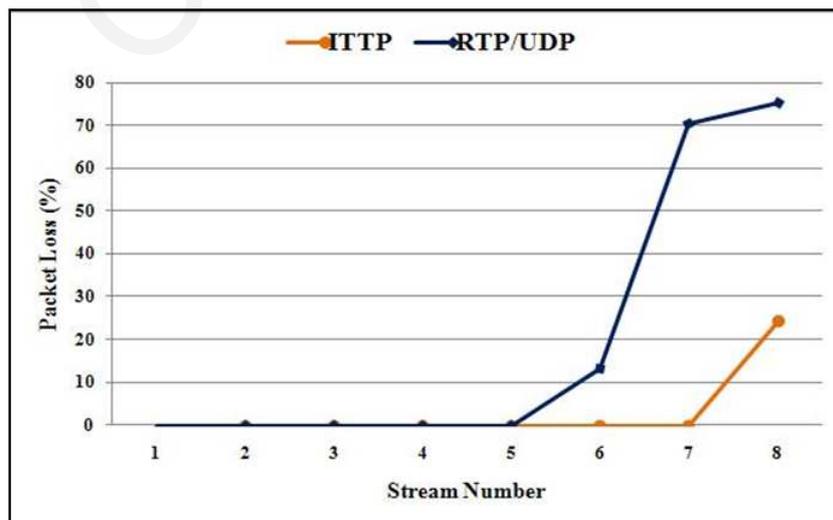


FIGURE 5. Packet loss ratio for ITTP and RTP/UDP

6.1.3. *Experiment 2: Delay.* In this experiment, two different delay cases of ITTP and RTP/UDP were investigated at various stream numbers between 1 and 10. In the first case, all the links had a bandwidth of 240 kbps. Figure 6 depicts the delay of both ITTP and RTP/UDP. The delay for ITTP is less than that for RTP/UDP under similar conditions. In addition, the difference in delay increases with the number of streams. For example, the delay of the first ITTP stream is 46.13 ms, whereas that of the first RTP/UDP stream is 48 ms. Therefore, the difference in delay is 1.87 ms. On the other hand, the delay of the tenth ITTP stream is 59.93 ms, whereas that of the tenth RTP/UDP stream is 66 ms. Therefore, the difference in delay is 6.07 ms.

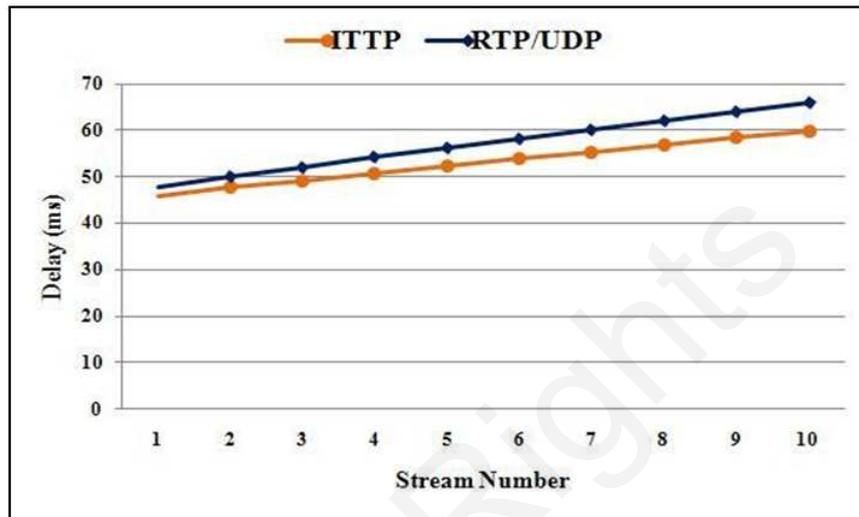


FIGURE 6. Delay case 1 for ITTP and RTP/UDP

In the second case, all links had a bandwidth of 227 kbps. Figure 7 depicts the delay of both ITTP and RTP/UDP. The delay of ITTP is less than that of RTP/UDP under similar conditions. In addition, the difference in delay increases with the number of streams. For example, the delay of the first ITTP stream is 46.49 ms, whereas that of the first RTP/UDP stream is 73.66 ms. Therefore, the difference in delay is 27.17 ms. On the other hand, the delay of the 10th ITTP stream is 61.06 ms, whereas that of the 10th RTP/UDP stream is 92.69 ms. Therefore, the difference in delay is 31.06 ms.

For both cases, the delay caused by ITTP was less than that caused by RTP/UDP. This was because the transmission delay, the processing delay, and the queuing delay of RTP/UDP were more than those of ITTP, as explained in Section 5.2.1.

The delay in the second case was greater than that in the first case. This is because the 240 kbps bandwidth in the first case was enough for 10 RTP/UDP streams, whereas the 227 kbps bandwidth in the second case was not enough. Therefore, the queuing delay in the second case will be much longer than that in the first case. On the other hand, the bandwidth was enough for 10 ITTP streams for both cases. Therefore, the queuing delay was approximately zero.

6.1.4. *Experiment 3: Number of calls supported.* In this experiment, the number of concurrent calls that can be run at various bandwidths between 100 and 500 kbps was explored. For each different link bandwidth, the number of concurrent calls was increased. When packet loss starts, the link is considered overloaded. Therefore, the number of concurrent calls for each link bandwidth is equivalent to the number of calls before the packet loss started. Figure 8 depicts the number of concurrent calls for both ITTP and

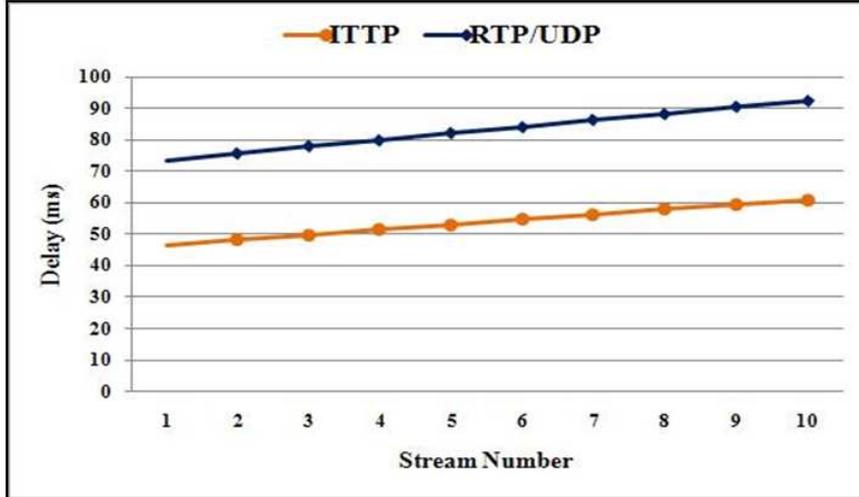


FIGURE 7. Delay case 2 for ITTP and RTP/UDP

RTP/UDP. The number of concurrent calls when using ITTP is more than the number of concurrent calls when using RTP/UDP under the same channel bandwidth. This is because the RTP/UDP stream consumes more bandwidth than the ITTP stream. Moreover, the difference in the number of concurrent calls increased as the channel bandwidth was increased.

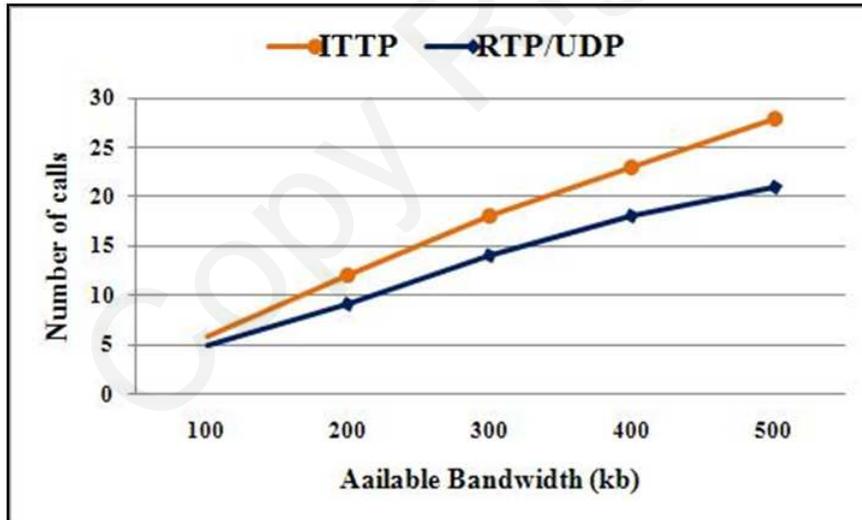


FIGURE 8. Number of calls supported by ITTP and RTP/UDP

6.1.5. *Experiment 4: Goodput.* Throughput is the amount of data delivered to the destination at a specific time, including the header, acknowledgment, and retransmitted packet data. However, using the throughput to evaluate ITTP performance will not yield accurate results because it calculates both useful and non-useful data. On the other hand, goodput is the amount of packet payload delivered to the destination. Unlike throughput, the goodput is an accurate evaluation for the protocol where only the useful data are calculated. Equation 5 is used to calculate the goodput:

$$Goodput = \frac{\sum R_{Pkt} * Pkt_s * 8}{1000} \quad (5)$$

where  $R_{Pkt}$  is the received packet, and  $Pkt_s$  is the received packet payload size. In this experiment, the goodput was used to evaluate ITTP compared with RTP/UDP at various bandwidths between 100 and 350 kbps. Figure 9 depicts the goodput of ITTP and RTP/UDP. Better bandwidth utilization was observed when using ITTP compared with RTP/UDP. This is because the available bandwidth is used to transfer the actual voice data when using ITTP, whereas most of the available bandwidth is used to transfer the header data when using RTP/UDP.

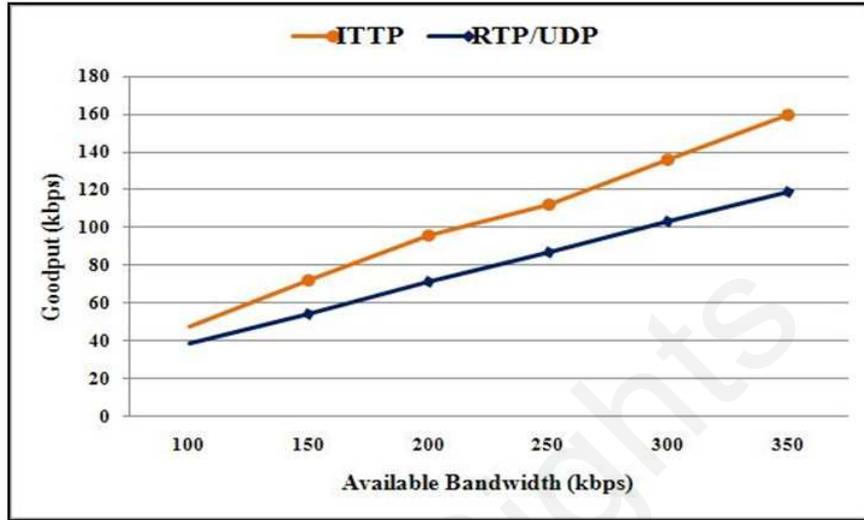


FIGURE 9. Goodput for ITTP and RTP/UDP

**6.2. Topology 2: Long-distance calls.** Topology 2 was utilized to evaluate ITTP performance for long-distance calls, where the distance between the call endpoints was six hops. Two different experiments with different parameters were run on Topology 2. In the first experiment, the ITTP packet loss was investigated and compared with that of RTP/UDP. In the second experiment, the ITTP delay was investigated and compared with that of RTP/UDP. In Topology 2, the parameters for the streams in each experiment vary. Therefore, the results increase or decrease variably, as the results will show.

**6.2.1. Topology 2 setup.** Topology 2 consists of 10 PCs and 6 routers. All LAN links between the PCs and the routers have a delay of 5 ms, all WAN links between the routers have a delay of 20 ms, and each node uses a DropTail queue, of which the maximum size is 50. A CBR traffic generator is attached to each PC. The data rates when using RTP/UDP and ITTP are 24 and 18.4 kbps, respectively. The rate varies because of the different header sizes. Figure 10 depicts Topology 2.

**6.2.2. Experiment 1: Packet loss.** In this experiment, the packet loss of ITTP and RTP/UDP was investigated at various stream numbers between 1 and 5. All links have a bandwidth of 60 kbps. PC1, PC2, PC3, PC4 and PC5 start transmitting data at 0.05, 0.16, 0.275, 0.385 and 0.491 seconds, respectively, and stop transmitting the data at 1.05, 1.16, 1.275, 1.385, and 1.491 seconds, respectively. Figure 11 depicts the packet loss ratio of both ITTP and RTP/UDP. The packet loss for ITTP is less than that for RTP/UDP under similar conditions. This is due to two main reasons. First, RTP/UDP causes a higher packet overhead than ITTP. Therefore, the RTP/UDP packet size is bigger than the ITTP packet size. The intermediate router buffer can also store more ITTP packets than RTP/UDP packets. As a result, the packet loss is decreased compared with

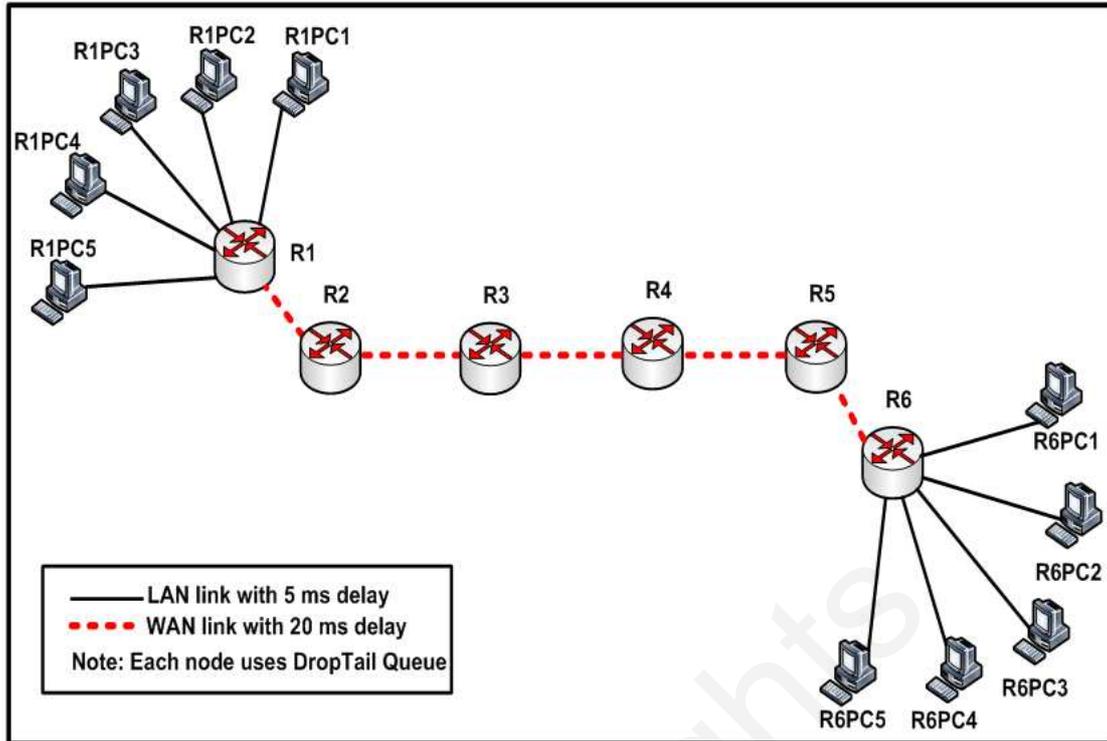


FIGURE 10. Topology 2

RTP/UDP. Second, the processing time for ITTP packets is less than that for RTP/UDP packets, thereby decreasing the packet loss as well.

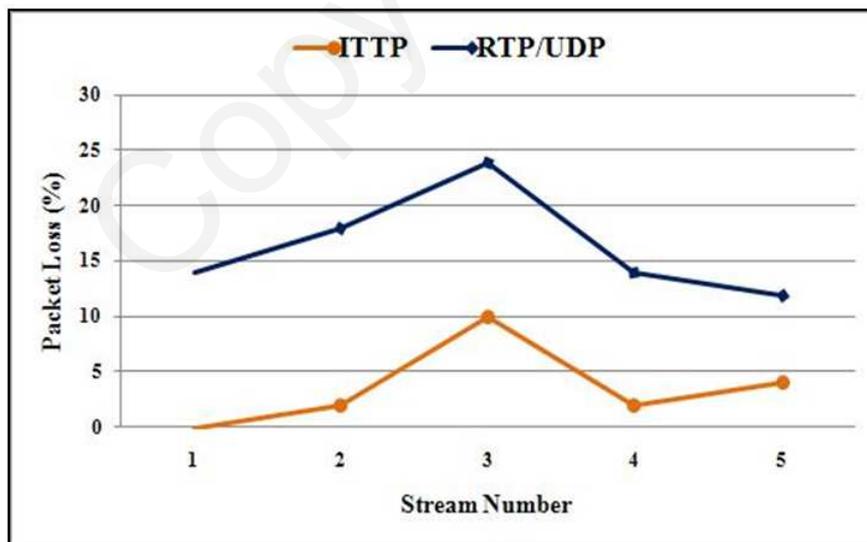


FIGURE 11. Packet loss ratio for ITTP and RTP/UDP

6.2.3. *Experiment 2: Delay.* In this experiment, two different delay cases of ITTP and RTP/UDP were investigated at various stream numbers between 1 and 5. PC1, PC2, PC3, PC4 and PC5 start transmitting data at 0.05, 0.06, 0.07, 0.08 and 0.09 seconds and stop transmitting data at 1.1, 1.16, 1.22, 1.28 and 1.34 seconds, respectively. In the first case, all links have a bandwidth of 120 kbps. Figure 12 depicts the delay of both

ITTP and RTP/UDP. The delay for ITTP is less than that for RTP/UDP under similar conditions.

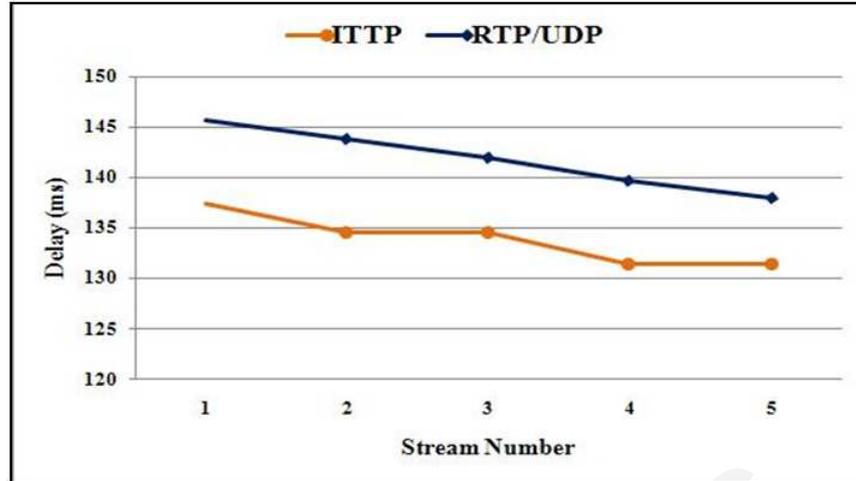


FIGURE 12. Delay case 1 for ITTP and RTP/UDP

In the second case, all links have a bandwidth of 105 kbps. Figure 13 depicts the delay of both ITTP and RTP/UDP. The delay for ITTP is less than that for RTP/UDP under similar conditions.

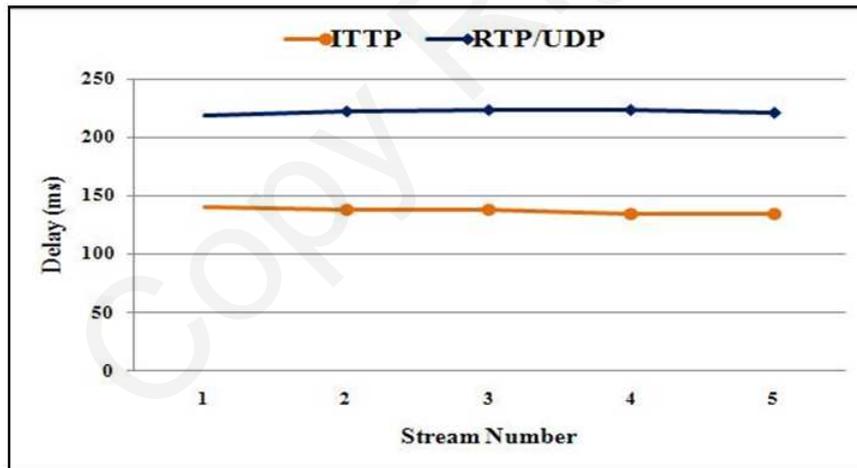


FIGURE 13. Delay case 2 for ITTP and RTP/UDP

In the two cases, the delay caused by ITTP was less than that caused by RTP/UDP. This was because the transmission delay, the processing delay, and the queuing delay of RTP/UDP were greater than those of ITTP, as explained in Section 5.2.1.

The delay in the second case was greater than that in the first case. This is because the 120 kbps bandwidth in the first case was enough for 5 RTP/UDP streams, whereas the 105 kbps bandwidth in the second case was not. Therefore, the queuing delay in the second case will be greater than that in the first case. On the other hand, the bandwidth was enough for 5 ITTP streams in both cases. Therefore, the queuing delay was approximately zero.

In conclusion, theoretical, mathematical, and simulation evaluations have been conducted in the current paper to evaluate ITTP. The obtained results are very relevant to

the protocol as its application is challenged by a number of limitations and difficulties at its early stage of development. However, the obtained results give both researchers and developers a better understanding of the value and the importance of this new VoIP protocol. Furthermore, they provide relevant feedback at this stage of protocol development so that it can be implemented in an error-free way.

**7. Conclusion.** The propagation of VoIP technology is hindered by problems caused by inefficient bandwidth usage and voice quality degradation. VoIP technology uses existing transport protocols, typically UDP, in conjunction with RTP to transfer VoIP data.

In the current paper, a new transport protocol called ITTP, which is dedicated to carrying VoIP application data, has been designed. ITTP addresses only the key functions of VoIP applications. ITTP has reduced header overhead compared with RTP/UDP, the currently used protocols. When using an 8 kbps codec with a 20 ms packetization and 20 byte packet size, ITTP adds a 30% overhead, whereas RTP/UDP adds a 100% overhead.

On one hand, the mathematical model and the simulation results have showed that ITTP utilizes the available bandwidth and increases the number of concurrent calls more effectively compared with RTP/UDP. The packet loss and delay when using ITTP is also less than those when using RTP/UDP. Therefore, both traffic handling and voice quality are better when using ITTP.

## REFERENCES

- [1] H. M. Chong and H. S. Matthews, Comparative analysis of traditional telephone and voice-over-Internet protocol (VoIP) systems, *2004 IEEE International Symposium on Electronics and the Environment*, 2004.
- [2] X. Wang, J. Lin, Y. Sun, H. Gan and L. Yao, Applying feature extraction of speech recognition on voip auditing, *International Journal of Innovative Computing, Information and Control*, vol.5, no.7, pp.1851-1858, 2009.
- [3] M. AbuAlhaj, M. S. Kolhar, M. Halaiyqah, O. Abouabdalla and R. Sureswaran, Multiplexing SIP applications voice packets between SWVG gateways, *Proc. of the 2009 International Conference on Computer Engineering and Applications*, pp.237-241, 2009.
- [4] T. Abbasi, S. Prasad, N. Seddigh and I. Lambadaris, A comparative study of the SIP and IAX VoIP protocols, *Canadian Conference on Electrical and Computer Engineering*, 2005.
- [5] J.-G. Doong and C.-S. D. Wei, Secured operation planning on service networks, *International Journal of Innovative Computing, Information and Control*, vol.5, no.3, pp.609-618, 2009.
- [6] *Packetizer, H.323 versus SIP: A Comparison*, [http://www.packetizer.com/ipmc/h323\\_vs\\_sip/](http://www.packetizer.com/ipmc/h323_vs_sip/), 2011.
- [7] M. M. Abu-Alhaj, M. S. Kolhar, L. V. Chandra, O. Abouabdalla and A. M. Manasrah, Delta-multiplexing: A novel technique to improve VoIP bandwidth utilization between VoIP gateways, *2010 IEEE 10th International Conference on Computer and Information Technology*, 2010.
- [8] T. Hoshi, K. Tanigawa and K. Tsukada, Proposal of a method of voice stream multiplexing for IP telephony systems, *Internet Workshop*, 1999.
- [9] C. Bouras, A. Gkamas and G. Kioumourtzis, Extending the functionality of RTP/RTCP implementation in network simulator (NS-2) to support TCP friendly congestion control, *Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering*, 2008.
- [10] M. Boucadair, *Inter-Asterisk Exchange (IAX): Deployment Scenarios in SIP-Enabled Networks*, 1st Edition, John Wiley & Sons Ltd., 2009.
- [11] M. Abu-Alhaj, M. S. Kolhar, M. Halaiyqah, O. Abouabdalla and R. Sureswaran, MuxComp – A new architecture to improve VoIP bandwidth utilization, *2009 International Conference on Future Networks*, 2009.
- [12] H. Schulzrinne and S. Casner, *RFC3551: RTP Profile for Audio and Video Conferences with Minimal Control*, RFC Editor, United States, 2003.
- [13] C. Perkins, *RTP: Audio and Video for the Internet*, Addison-Wesley Professional, 2003.
- [14] M. Spencer, B. Capouch, E. Guy, F. Miller and K. Shumard, *Inter-Asterisk eXchange*, 2nd Edition, IETF RFC 5456, 2010.

- [15] M. SK, M. Abu-Alhaj, O. Abouabdalla, T.-C. Wan, R. Budiarto and A. M. Manasrah, Conference gateway for heterogeneous clients: Real time switching clients and interasterisk exchange clients, *International Journal of Innovative Computing, Information and Control*, vol.7, no.1, pp.395-406, 2011.
- [16] K. Egevang and P. Francis, *RFC1631: The IP Network Address Translator*, RFC Editor, United States, 1994.
- [17] C. M. Kozirook, *The TCP/IP Guide*, 3rd Edition, 2005.
- [18] B. Goode, Voice over Internet protocol (VoIP), *Proc. of the IEEE*, vol.90, no.9, pp.1495-1517, 2002.
- [19] L.-Å. Larzon, M. Degermark and S. Pink, *UDP Lite for Real Time Multimedia Applications*, Citeseer, 1999.
- [20] S. Casner and V. Jacobson, Compressing IP/UDP/RTP headers for low-speed serial links, *RFC 2508*, 1999.
- [21] H. P. Sze, S. C. Liew, J. Y. B. Lee and D. C. S. Yip, A multiplexing scheme for H.323 voice-over-IP applications, *IEEE Journal on Selected Areas in Communications*, vol.20, no.7, pp.1360-1368, 2002.
- [22] N. Moghim, S. M. Safavi and M. R. Hashemi, Performance evaluation of a new end-point admission control algorithm in NGN with improved network utilization, *International Journal of Innovative Computing, Information and Control*, vol.6, no.7, pp.3067-3080, 2010.
- [23] A. Ito, T. Sakai, K. Konno and S. Makino, Packet loss concealment for mdct-based audio codec using correlation-based side information, *International Journal of Innovative Computing, Information and Control*, vol.6, no.3(B), pp.1347-1361, 2010.
- [24] B. Subbia, S. Sengodan and J. Rajahalme, RTP payload multiplexing between IP telephony gateways, *Global Telecommunications Conference*, 1999.
- [25] O. Hagsand, I. Marsh and K. Hanson, Sicsophone: A low-delay internet telephony tool, *IEEE*, 2003.
- [26] *ITU-T SG13, the Insufficiency of RTP for TDM Clock Recovery*, <http://www.dspscsp.com/tdmoip/index.html>, 2003.
- [27] K. James and R. Keith, *Computer Networking: A Top-down Approach*, Addison-Wesley, 2010.
- [28] S. Shin and H. Schulzrinne, Measurement and analysis of the VoIP capacity in IEEE 802.11 WLAN, *IEEE Transactions on Mobile Computing*, vol.8, no.9, pp.1265-1279, 2009.