TCP-over-UDP for Real Time Applications

Saurabh Tripathi, Shaurya Gupta, Tushar Babbar, Vishal Gupta

Abstract— Real Time Applications are getting more prevalent over internet and they require fast and reliable flow of information across the connected devices. The two major TCP/IP transport protocols fail to completely provide the desired feature for Real Time Applications to communicate. Thus many efforts have been made to provide a suitable protocol that can provide flow of real time data across the internet efficiently. Many protocols have been developed that try to provide TCP like functionalities to UDP in order to meet the needs in various networking scenarios.The main aim of this paper is to highlight a general solution based on these different efforts which can be used to achieve the desired functionality over internet for the Real Time Applications to communicate. This paper aims to present a generic library called as TCP-over-UDP for Real Time Applications based on [1]. This library can be optimised and refined in order to make it suitable for communication in real time environments.

Index Terms— RTA - Real Time Applications, RTC - Real Time Communication, RTP - Real Time Protocol, RUDP - Reliable User Datagram Protocol, RTMFP – Real Time Media Flow Protocol, TCP – Transmission Control Protocol, UDP – User Datagram Protocol

1 INTRODUCTION

A real-time application (RTA) is an application program that functions within a time frame that the user senses as immediate or current. The latency must be less than a defined value, usually measured in seconds. Such application may have communication requirements when used in distributed environment. In such a distributed environment these application adhere to Real Time Communication(RTC) guidelines. Real-time communications is any mode of telecommunications in which all users can exchange information instantly or with negligible latency. In this context, the term "real-time" is synonymous with "live."

In TCP/IP(internet) network communication among the different nodes of the network is based on the either of the two prevalent protocols which are Transmission control protocol(TCP) and User Datagram Protocol(UDP). But both of these protocols fails to provide the necessary requirements for the Real Time Applications to communicate. Thus the main purpose of this paper is to provide a solution to the current problem , TCP-over-UDP for Real Time Applications based on the better features of both TCP and UDP for Real Time Communication to be feasible to be established in the TCP/IP environment.

Many protocols and library implementations have talked about adding TCP like Features to UDP in order to make it efficient yet a faster means for communication and also to provide TCP like scenario where a TCP connection can not be made but a UDP can be [1]. Real Time transport protocols like Real Time Protocol (RTP -RFC 3550), Relaible UDP(RUDP), Real Time Media Flow Protocol (RTMFP) etc. describe the use of UDP protocol along with some additional capabilities can be used to provide a necessary scenario for the applications to communicate and exchange data in a real time scenario [2], [3], [4]. It is also been observed in [5] that TCP-over-UDP can be a faster option for communication to occur.

- *Tushar Babbar is currently pursuing graduation degree program in computer science engineering in Inderprastha Engineering College, India, PH-01204535007. E-mail: babbartushar@gmail.com*
- *Saurabh Tripathi is currently pursuing graduation degree program in computer science engineering in Inderprastha Engineering College, India, PH-01204535007. E-mail: saurabh.daksh91@gmail.com*
- *Shaurya Gupta is currently pursuing graduation degree program in computer science engineering in Inderprastha Engineering College, India, PH-01204535007. E-mail: tech.shaurya@gmail.com*
- *Vishal Gupta is currently pursuing graduation degree program in computer science engineering in Inderprastha Engineering College, India, PH-01204535007. E-mail: er.vishal92@gmail.com*

1.1 Real Time Applications

A real-time application (RTA) is an application program that functions within a time frame that the user senses as immediate or current. The latency must be less than a defined value, usually measured in seconds. Whether or not a given application qualifies as an RTA depends on the worst-case execution time (WCET), the maximum length of time a defined task or set of tasks requires on a given hardware platform. [6]
Examples of RTAs include:

- Videoconference applications
- VoIP (voice over Internet Protocol)
- Online gaming
- Community storage solutions
- Some e-commerce transactions
- Chatting
- IM (instant messaging)

1.2 Real Time communications

Real-time communications (RTC) is any mode of telecommunications in which all users can exchange information instantly or with negligible latency. In this

context, the term "real-time" is synonymous with "live."

RTC can take place in half-duplex or full-duplex modes. In half-duplex RTC, data can be transmitted in both directions on a single carrier or circuit but not at the same time. In full-duplex RTC, data can be transmitted in both directions simultaneously on a single carrier or circuit. RTC generally refers to peer-to-peer communications, not broadcast or multicast.

In RTC, there is always a direct path between the source and the destination. Although the link might contain several intermediate nodes, the data goes from source to destination without having to be stored anyplace. In contrast, timeshifting communications always involves some form of data storage between the source and the destination.[6]

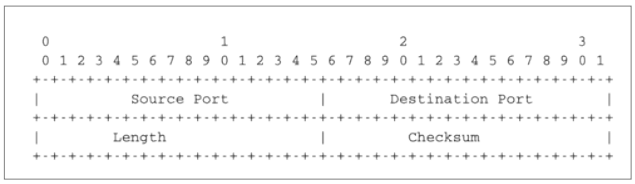
Requirements for Real time communication:

- Timeliness
- Flexibility
- Physical Performance Limitations
- Flowcontrol

1.3 User Datagram Protocol (UDP)

UDP uses a simple transmission model without implicit hand-shaking dialogues for guaranteeing reliability, ordering, or data integrity. Thus, UDP provides an unreliable service and datagram may arrive out of order, appear duplicate, or go missing without awareness of hosts. UDP assumes that the error checking and correction is either not necessary or performed in the application, avoiding the overhead of such processing at the network transport level.[9]

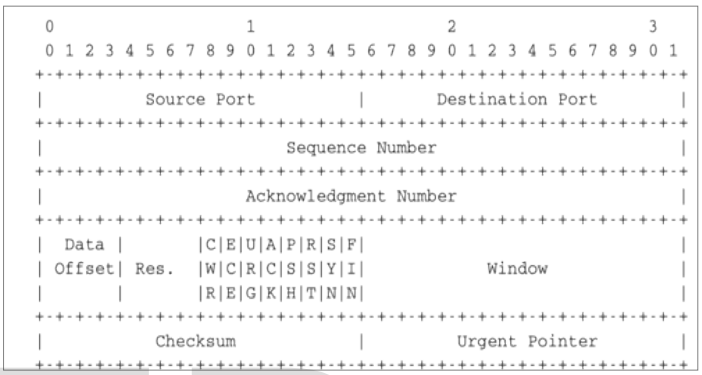
UDP header:



1.4 Transmission Control Protocol(TCP)

TCP provides a connection-oriented, reliable, and byte stream service. Firstly, the term connection-oriented means the two applications using TCP must establish a TCP connection before they can exchange data. Secondly, for achieving reliability, TCP assigns a sequence number to each byte transmitted, and expects a positive acknowledgment (ACK) from the receiving TCP. If the ACK is not received within a timeout interval, the data is retransmitted. The receiving TCP uses the sequence numbers to rearrange the segments when they arrive out of order, and to eliminate duplicate segments. Finally, TCP transfers a continuous stream of bytes. TCP does this by grouping the bytes in TCP segments, which are passed to IP for transmission to the destination. TCP itself decides how to segment the data and it may forward the data at its own convenience. In addition to the properties above, TCP is also a full duplex protocol, meaning that each TCP connection supports a pair of byte streams, one flowing in each direction.[8]

TCP header :



2. MOTIVATION TO BUILD TCP-OVER-UDP FOR REAL TIME APPLICATION

From the explanation above it is explainable that both TCP and UDP are incapable of providing the communication overheads of the Real Time Communication, thus it is not feasible for the Real Time Applications to be able to use any of the protocol mentioned above.

UDP being a connectionless unreliable service is fast, which is required for the RTC but it being unreliable makes it unusable in Real Time Environment. On the other hand TCP provides Reliability and Flow Control but being slow and expensive to use it can not be used in RTC. But, Mixing the better features of both the Protocols is can provide a solution which is fast and reliable. This is the main aim of this paper to highlight the need of such a solution.

It is explained in TCP-over-UDP draft[1] that a TCP instance can be made on top of UDP which provide exactly same functionality as TCP. It provides exactly the same congestion control, flow control, reliability, and extension mechanisms as offered by TCP. It is intended for use in scenarios where applications running on two hosts may not be able to establish a direct TCP connection but are able to exchange UDP packets.

Furthermore, [3] explains that UDP can be made reliable in the sense to make sure that the delivered packet make it to the desired destination. Also, [2], [4], contains the explanation of how UDP can be made suitable to be fit in the real time environment so that Real Time Applications can communicate in the internet scenarios. RTP [2] is carried on top of IP and UDP, and follows the conventions established by the profile for audio and video. Adobe secure RTMFP [4] states that it includes independent parallel message flows which may have different delivery priorities, variable message reliability (from TCP-like full reliability to UDP-like best effort), multipoint congestion control, and built-in security. Thus TCP-over-UDP for Real Time Applications can be a suitable option for the developers to create real time network applications.

3. Implementation of TCP-over-UDP for Real Time Applications

TCP is a well structured protocol which does a lot of overhead to provide all the features like Connection Establishment, Flow Control, Congestion Control, Recovery, Stream-like flow etc. To achieve such abilities TCP has to suffer with transmission speed and processing resources required at the OS level. Whereas UDP just focus on transmitting data from one side to other. Thus UDP can be provided with additional functionalities of the TCP so that it can work as a reliable yet faster protocol. Thus UDP can be molded in the desired form in order to provide it all those features which are necessary for Real Time Communication to Take place. The idea is to implement the different routines that TCP follow in order to provide reliability, connection establishment etc. in a separate process that could run on top of the UDP process in the kernel stack to provide all those features required by RTC. This way the Real Time Application can communicate to other such applications in a faster and reliable way.

3.1 Architecture of TCP-over-UDP

The architecture for TCP-over-UDP has been highlighted in [7]. The TCP-over-UDP process can thus communicate with the UDP process running in the Kernel space to provide fast transmission and also provide reliability and other features on its own to the Real Time Application. This way the needs for the real titme communication be met by the TCP-over-UDP process.
The necessary information that must flow along the network, which is contained by the TCP header can be included in the UDP Datagram such that the UDP packet will contain in its data field the necessary information required by tht TCP-over-UDP process before any user data. This way the necessary information can be sent across the network.

3.2 Reliability in UDP

The Reliable UDP protocol[3] gives the guidelines to make UDP a reliable protocol which is necessary for the real time communication to take place. It states that :

- transport should provide reliable delivery up to a maximum number of retransmissions (i.e. avoid stale signaling messages).
- transport should provide in-order delivery, if out-order packet received then it will be stored in an efficient data structure like circular queue[7] or linked list and retrieved when required, this will reduce the number of retransmissions.
- transport should be a message based.
- transport should provide flow control mechanism.
- transport should have low overhead, high performance.
- characteristics of each virtual connection should be configurable (i.e. timers).
- transport should provide a keep-alive mechanism.
- transport should provide error detection.
- transport should provide for secure transmission.

3.3 Sessions in Real Time Communication

An association among a set of participants communicating with RTP. A participant may be involved in multiple RTP sessions at the same time. In a multimedia session, each medium is typically carried in a separate RTP session with its own RTCP(Real Time Control Protocol) packets unless the the encoding itself multiplexes multiple media into a single data stream. A participant distinguishes multiple RTP sessions by reception of different sessions using different pairs of destination transport addresses, where a pair of transport addresses comprises one network address plus a pair of ports for RTP and RTCP. All participants in an RTP session may share a common destination transport address pair, as in the case of IP multicast, or the pairs may be different for each participant, as in the case of individual unicast network addresses and port pairs.

4. CONCLUSION

Implementing a library that can provide such an implementation can be useful in different scenarios where both speed of transmission and reliability is required. Such a library can be

used as base for Real Time Application developers to redesign the protocol in the way that suits their application. Such a library would have great implementation in different media applications which require audio and video transmission.

REFERENCES

[1] [draft-baset-tsvwg-tcp-over-udp-01] Salman A. Baset and Henning Schulzrinne, "TCP-over-UDP", baset-tsvwg-tcp-over-udp-01 (work in progress), Dec., 2009.

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

[3] [draft-ietf-sigtran-reliable-udp-00] T. Bova, T. Krivoruchka, "Reliable UDP Prototcol", 25 Feburary 1999.

[4] [draft-thornburgh-adobe-rtmfp-07] M. Thornburgh , "Real Time Media Flow Protocol", May 2, 2013.

[5] Dunigan, T. and F. Fowler, "Almost TCP over UDP", 2004, <http://www.csm.ornl.gov/~dunigan/netperf/atou.html >.

[6] Hermann Kopetz, "Real-Time Systems: Design Principles for Distributed Embedded Applications", 2011.

[7] Cheng-HanLee, Salman Abdul Baset, ChinmayGaikwad, <www.cs.columbia.edu/~cl2804/projects/CU_ToU_2009/index.html>.

[8] [RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, September 1981.

[9] [RFC0768] Postel, J., "User Datagram Protocol", ISI, RFC 768, 28 August 1980.