Volume 3, No. 5, Sept-Oct 2012

ISSN No. 0976-5697

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Method to Reduce the Overflow of Rtcp Packets While Transmitting Multimedia Components over Internet

Meenu Vijarania
Lecturer, Department of Computer Science
Amity University Haryana Gurgaon, India
meenuhans.83@gmail.com

Abstract: This paper presents the design and performance analysis of Real-time Transport Control Protocol. RTP, the Real Time Transport Protocol, has been used as the transport protocol for voice and video on the Internet. Its companion control protocol, is the Real Time Control Protocol (RTCP). Together these protocols provide controlled delivery of multimedia traffic over the Internet. RTCP is used for loose session control, QoS reporting, and media synchronization, delay, jitter and packet loss calculations. The RTP specification describes an algorithm for determining the RTCP packet transmission rate at a host participating in a multicast RTP session. This algorithm was designed to allow RTP to be used in sessions with anywhere from one to a million members. However, we have discovered several problems with this algorithm when used with very large groups with rapidly changing group membership. One problem is the flood of RTCP packets which occurs when many users join a multicast RTP session at nearly the same time. To solve this problem, this paper proposed a novel adaptive timer algorithm called timer revaluation algorithm. This paper demonstrate that it performs extremely well, reducing the congestion problem by several orders of magnitude.

Keywords: RTCP, QoS, RTP, NTP, SR, RR

I. INTRODUCTION

Multimedia services, such as video conferencing, Internet telephony and streaming audio, have recently been introduced for the millions of users of the Internet. The popularity of these services and the feedback received has clearly revealed that some modifications and extensions to the current internet protocols are needed to be able to support real-time applications better. Minimization of the end-to-end delay, accurate synchronization of the voice and video streams and a feedback mechanism for the quality of service monitoring are some of the main requirements of these various multimedia applications[1]. In the Internet, large multimedia sessions typically rely on the Real-time Transport Protocol (RTP) for data transmission and on its accompanying Real-time Transport Control Protocol (RTCP) for the distribution of feedback and control information [2] . Both RTP and RTCP originally assumed the availability of a shared multicast channel, allowing for efficient many-to-many communication among session participants any group member could communicate with all others by simply sending to a multicast address and the routing-level architecture would efficiently distribute the data.

The primary function of RTCP is periodically transmit sender and receiver reception reports to all participants in the session in order to facilitate session debugging[2]. When examined in the aggregate, reception reports allow a host to determine if a problem exists in the quality of the data distribution and if the problem is local or global. Used with multicast, RTCP allows third-parties not necessarily involved in the session to monitor reception quality and to diagnose network problems. RTCP is also used for inter-stream synchronization (e.g., when there are separate audio and video streams), as well as for the distribution of minimal control information, such as transport- level identifiers to track session participants. In addition, RTCP provides a means for group participants to estimate session size[3]. As a control protocol, RTCP must be sensitive to the amount of traffic it generates, so as not to interfere with the accompanying RTP real-time data streams. Normally, control bandwidth is dynamically determined based upon the proportion of active senders to receivers. The principle difficulty in achieving scalability to large group sizes is the rate of RTCP packet transmissions from a host. If each host sends packets at some fixed interval, the total packet rate sent to the multicast group increases linearly with the group size, ***N.*** This traffic would quickly congest the network [4].

To counter this, the RTP specification requires that end systems utilizing RTP listen to the multicast group, and count the number of distinct RTP end systems which have sent an RTCP packet. This results in a group size estimate, *L*, computed locally at each host. The interval between packet transmissions is then set to scale linearly with *L*. This has the effect of giving each group member (independent of group size) a fair share of some fixed RTCP packet rate to the multicast group. The flood of packets caused by the current RTCP algorithm with a step join has both good and bad consequences. The rapid arrival of RTCP packets causes a quick convergence to the correct group size estimate, which is good. However, the real packets of interest are the RTP media packets, not the RTCP packets [1]. Because of the restricted amount of bandwidth available at many access links, we believe that maintaining the RTCP rate at 5% of the session bandwidth is the goal of any fix for the flooding problem[4].

II. REAL TIME TRANSPORT CONTROL PROTOCOL

The RTP data transport is augmented by a control protocol (RTCP), which provides the RTP session participants feedback on the quality of the data distribution. The underlying protocol must provide multiplexing of the data and control packets, With UDP this is usually implemented using separate port numbers[2]. The format of the RTCP packets is fairly Similar to RTP packets, e.g. the type Indication is at the same location . The main function of the RTCP are:

a. QoS monitoring and congestion control
b. Identification
c. Session size estimation and scaling

The RTCP packets contain direct information for quality-of-service monitoring. The sender reports (SR) and receiver reports (RR) exchange information on packet Losses, delay and delay jitter. This information may be used to implement a TCP like flow control mechanism upon UDP at the application level using Adaptive encodings. A network management tool may monitor the network load based on the RTCP packets without receiving the actual data or detect the faulty parts of the network[1]. The RTCP packets carry also a transport level identifier (called a canonical name) for a RTP source, which is used to keep track of each participant. Source description packets may also contain other textual information (user's name, email address) about the source. Albeit the source of the RTP packets is already identified by the SSRC identifier, an application may use multiple RTP streams, which can be easily associated with this textual information. The RTCP packets are sent periodically by each session member in multicast fashion to the other participants[1]. The more there are participants the more RTCP messages should be exchanged. That's why the fraction of the control traffic must be limited. There is in fact a trade-off between up-to-date information and the amount of the control traffic. The control traffic load is scaled with the data traffic load so that it makes up about 5% of the total data traffic. There are, however, some weaknesses related to the scalability of the current RTCP algorithms[1]. These problems are listed in below.

a. Congestion due to floods of RTCP packets in highly dynamic groups.
b. Large delays between receipt of RTCP packets from a single user.
c. Large size of the group membership tables.

A. RTCP Packet Formats:

Each RTCP packet starts with an header similar to that of the RTP data packets. The payload type field identifies the type of the packet. In [5] there are five RTCP payload types (200-204) defined:

a. Sender Report (SR)
b. Receiver Report (RR)
c. Source Description (SDES)
d. Goodbye (BYE)
e. Application-defined packet (APP)

The contents of these packets are in detail described in Figure 1.

V	P	RC	PT=200	Length
SSRC of the sender				
NTP timestamp (MSB)				
NTP timestamp (LSB)				
RTP timestamp				
Sender's packet count				
Sender's octet count				
First reception report block (SSRC_1)				
...				
Last reception report block (SSRC_n)				

Figure 1: Format of the Sender Report [2]

The first 32 bits of the header of the sender report consists of several control bits. The version number (V) and padding field (P) are the same as in RTP packet.The reception report count (RC) indicates the number of receiver reports attached to this packet. The maximum number of receiver reports is 32. The payload type (PT) for sender report is 200. The length field indicates the length of the packet in 32-bit words minus one. The second 32-bit word includes the SSRC of the sender and the next two words include the high and low parts of the 64-bit NTP (Network Time Protocol) timestamp. The RTP timestamp indicates the relative sending time of this packet. Last sender related words include the sender's packet and octet counts. Following the sender's information block there are zero or more reception report blocks, which follow the same format as in the receiver reports [1].

V	P	RC	PT=201	Length
SSRC of the sender				
SSRC of the first source				
Fract. lost			Cum. no of packets lost	
Ext. highest sequence number received				
Interarrival jitter estimate				
Last sender report timestamp (LSR)				
Delay since last sender report (DLSR)				
...				
Last reception report block				

Figure 2: Format of the Receiver Report [2]

The first 32-bit word in that block is the SSRC of the source, for which this reception report is aimed. The fraction lost field indicates the number of packets lost divided by the number of packets expected (according to the highest sequence number received) since last receiver report. The lower part of the next 32- bit word includes the highest sequence number received since last report, whereas the higher part is used as and extension to the sequence number revealing possible resets of the sequence numbering.

V	P	SC	PT=202	Length
SSRC/CSRC of the sender				
Type		length		text
text continued				
...				
Last chunk				

Figure 3: Format of the Source Description [2]

The Source Description (SDES) packet is three-level structure composed of a header and zero or more chunks (greyed area in the table 4), which describe the source identified in that particular chunk. An end system sends only one chunk with its SSRC but a mixer incorporates as many chunks as there are contributing sources to be identified. Each SDES item starts with an 8-bit type field followed by an 8-bit octet count, which identifies the length of the following text field. The defined SDES items are: canonical end-point identifier (CNAME), which should follow the format *user@host*, user name (NAME), being the real user name, electronic mail address (EMAIL) in format John.Doe@megacorp.com, phone number (PHONE), geographical user location (LOC), application or tool name (TOOL), notice (NOTE) and private extensions (PRIV). Only the item CNAME is mandatory.

V	P	SC	PT=203	Length
SSRC/CSRC of the sender				
length			reason for leaving	
...				
Last chunk				

Figure 4: Format of the BYE packet [2]

The BYE packet indicates the receivers that a source is leaving the session and the prolonged silence will be caused by that reason instead of a network failure. The BYE packet may optionally include a textual description of the reason for leaving.

III. PROPOSED ALGORITHM : TIMER REVALUATION

As mentioned previously, the current RTCP algorithm of scaling the transmission interval of the RTCP reports is linearly proportional to the group size estimate (L)[8]. As the group size grows, a sender and receiver reports are sent less frequently [9]. This algorithm works fine for group sizes up to several hundreds but when scaled to a very large and very dynamic multicast group certain problems may arise. It can be observed that in large multicast groups, in cabel TV networks for example, a great number of users change channels at almost the same time when shows begin and end. This "step-join" phenomenon is not handled very efficiently with the current RTCP algorithm [4]. The unrestricted flood of RTCP packets in case of large step-join is very likely to cause congestion, which even makes the situation worse because disappeared packets keep the group size estimates inaccurate. In these situations the 5% target for control traffic is most likely exceeded. A timer revaluation method is proposed, which should restrict the number of packets sent especially in rapid step-join environments. The Timer Revaluation algorithm computes for transmission interval for RTCP packets using equation(1).

$$\mathbf{t_n = (t_{n-1} + 2*t_{n-2})t_{n-1} + R(a)\max(T_{min}, CL(t_{n-1}))} \qquad \textbf{(1)}$$

Where t_n is the current sending time, t_{n-1} is the previous sending time, R(a) is a randomizing factor between 0.5 and 1.5, T_{min} is initially 2.5s and 5s after that, C is a priori calculated interval according to 5% target for the control bandwidth and $L(t_{n-1})$ is the previous group size estimate.

In practice, at time t_{n-1} a timer is set to be run out at time t_n for sending the next packet. The timer revaluation algorithm changes this scheme so that when timer has run out the sending time is recalculated using the most recent information about the current group size. The group size estimate $L(t_n)$ may have a ready changed rapidly from t_{n-1} to t_{n-1} in case of a large step-join. If the recalculated sending time is beyond the initial t_n, the packet is rescheduled to be sent later. Otherwise it is sent according to the initial plan. Two operation modes for timer revaluation algorithm are proposed: **conditional** and **unconditional** [5].

With conditional mode the timer revaluation is done only if group size estimate has changed. With unconditional mode the timer revaluation is always done, which makes the timer revaluation to act more raidly when the group size changes beacause incoming reports are not waited. Also the randomization smoothes the beginning of the group size increase. The presents a simulation results of timer revaluation algorithm seen by a single user when 10000 new participants join the session. The step-join causes a burst of 10000 packets which are sent in current algorithm to be reduced to 197 packets with conditional and to 75 packets with unconditional timer revaluation. These values are far more close to 5 % target of RTCP traffic than that of all sending initally at full speed.

IV. RESULT AND ANALYSIS

In our analysis I have used a network simulator called NS2 that works with the following characteristics:

a. The packet size equals 512 byte.
b. Session bandwidth equals 250 Kbps.
c. Delay time between each subsequent user RR equals one second.
d. The simulation is done with session size (10, 20,…) as a start.
e. The simulation time is 80 seconds.
f. The session size is dynamic.
g. The simulation can contain multiple sender case.
h. The timer and user number (inputs and outputs) are generated by a random number
i. generator function.
j. The packet formation time is neglected. It is supposed that the packet is formed during the delay time (one second).

A. *Bandwidth utilization evaluation:*

To analyze our scheme in relation to the bandwidth utilization, we determined the average number of feedback RRs which are sent during the simulation time. If the number of RRs increases with notable values. during any time interval, this may lead to decrease the bandwidth for the multimedia data. Consequently, the congestion problem probably occurs. The significant decrease in the RRs is justified by the event that a large number of old participants left the session and new participants are gradually joining to the session.

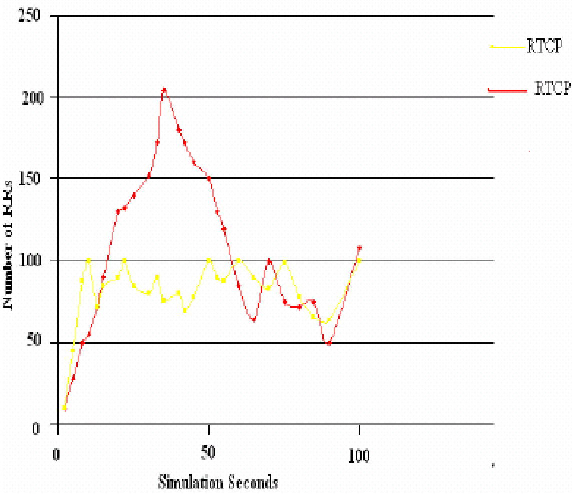

Figure 5: Bandwidth Utilization Evaluation

The Bandwidth utilization has increased by 17.4%

B. Interval calculated using timer revaluation algorithm:

Simulation have shown that the algorithm reduces the initial congestion by orders of magnitude under a variety of conditions.

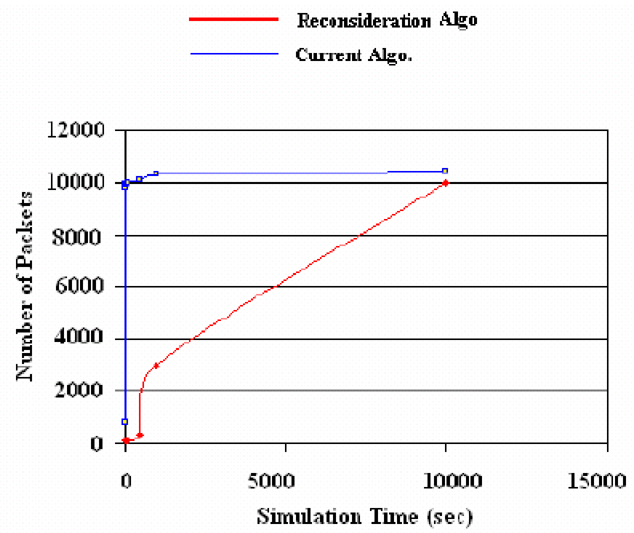

Figure 6: Congestion control comparision of two algorithms

V. CONCLUSION

RTP was meant to support real-time communications ranging from two-party telephone calls to broadcast applications with very large user populations. It incorporates an adaptive feedback mechanism that allows scaling to moderately sized groups, but shows a number of deficiencies once the group size exceeds on the order of a thousand. The problems can be summarized as congestion, convergence delays and state storage problems. I have proposed a solution for congestion problem via a simple algorithm called timer revaluation.

Simulations have shown that the algorithm reduces the initial congestion under a variety of conditions. Future work involves resolving the other RTP scalability problems: state storage.

VI. REFRENCES

[1]. Tommi Koistinen ,"Protocol overview: RTP and RTCP" Nokia telecommunications. http://www.netlab.tkk.fi/opetus/s38130/k99/presentations/4.pdf

[2]. H. Schulzrinne, S. Casner R. Frederick Request for Comments: 3550 Network Working Group Columbia University Blue Coat Systems Inc. V. Jacobson July 2003. http://www.ietf.org/rfc/rfc3550.txt

[3]. Julian Chesterfield,Eve M. Schooler " An Extensible RTCP Control Framework for Large Multimedia Distributions", Proceedings of the Second IEEE International Symposium on Network Computing and Applications 0-7695-1938-5/03,2003 IEEE

[4]. Jonathan Rosenberg, Henning Schulzrinne "Timer Revaluation for Enhanced RTP Scalability" IEEE Bell Laboratories Columbia University , 1998

[5]. H. Schulzrinne, S. Casner R. Frederick Request for Comments: 1889 and 1890 Network Working Group Columbia University Blue Coat Systems Inc. V. Jacobson, July 2003, http://www.ietf.org/rfc/rfc1890.txt, http://www.ietf.org/rfc/rfc1889.txt

[6]. Randa El-Marakby, David Hutchison Computing Dept., "Scalability Improvement of the Real-time Control Protocol (RTCP) Leading to Management Facilities in the Internet" IEEE Lancaster University

[7]. Busse,I.Deffner, B.Schultzrinne H. "Dynamic QoS Control of Multimedia Application based on RTP", Computer Communications January(1996).

[8]. Bernard Aboba, "Alternatives for enhancing RTCP scalability," Internet Draft, Internet Engineering Task Force, Jan.1997, Work in progress. http://tools.ietf.org/html/draft-aboba-rtpscale-02

[9]. Randa El-Marakby , David Hutchison, "Scalability improvement of the real time control protocol", Computer Communications, v.28 n.2, p.136-149, February, 2005

[10]. Randa El-Marakby, "Design and Performance of a Scalable Real Time Control Protocol: Simulations and Evaluations", Proceedings of the Fifth IEEE Symposium on Computers and Communications (ISCC 2000), p.119, July 04-06, 2000