Available online www.jocpr.com

Journal of Chemical and Pharmaceutical Research, 2014, 6(6):2298-2302

Research Article

ISSN : 0975-7384
CODEN(USA) : JCPRC5

TCP-like congestion control algorithm for stream media transmission

Xiaoyan Zhao* and Huili Meng

College of Computer and Information Engineering, Henan Normal University, Henan, China

---

ABSTRACT

Currently, the influx of streaming media business tends to aggravate network congestion. In order to improve network efficiency, a new TCP-like congestion control algorithm based on RTP/TRCP protocol model is proposed in this paper. This algorithm uses an improved AIMD mechanism to control window for adjusting the transmission rate of RTP data frame by adopting TCP acknowledgment mechanism and RTCP report. This algorithm firstly deduces the formula of average delay jitter and defines some parameters such as round trip time and timeout clock value. Secondly, a kind of transmission control strategy based on feedback is presented. More importantly, this algorithm completely abandon the retransmission mechanism of TCP to keep continuity and real-time of streaming media business. The experimental results show that, compared with other RTP flow control algorithm, the proposed algorithm has better congestion control ability and it can restrain the jitter effectively.

Keywords: Congestion Control; Real-time Transmission Protocol; Round Trip Time; Rate Control; Timeout Clock

---

INTRODUCTION

The influx of streaming media business makes network congestion increase. Compared with traditional businesses of HTTP and FTP which are based on TCP, the streaming media business has greater difference, as described in [1]. It is allowed to a certain degree of error and loss to ensure the real-time, strong continuity and large amount of data transmission. The business quality is very sensitive to variation of bandwidth, also has special needs to QoS, which is validated in [2]. Therefore, a congestion control method adapted to its characteristics to guarantee the performance and quality of transmission network is needed.

Congestion control algorithm, based on TCP, is a mature congestion control mechanism in current network. [3-4] proposed an additive increase multiplicative decrease (AIMD) window control mechanism based on TCP protocol; [5-6] used different TCP Reno slow start strategy to carry out congestion control. [7-10] presented a variety of new congestion control algorithm which adopt TCP friendly rate control (TFRC) strategy. However, in order to be compatible with traditional TCP business and ensure integrity of data, the improved TCP congestion control mechanism must use the mechanisms of fast recovery and timeout retransmission, etc, which seriously impacted the continuity and real time of streaming media and reduced the business quality. To solve these problems, a TCP-like streaming media congestion control algorithm based on RTP/RTCP protocol is proposed, which focuses on how to improve TCP congestion control mechanism and apply it to RTP/RTCP protocol, while enhancing the fairness of streaming media and reducing delay jitter to ensure transmission quality of streaming media business.

PRINCIPLE OF TCP-LIKE ALGORITHM

TCP-like algorithm is composed of receiving acknowledgment module, window control module, transmission control module and clock control module, etc. The size of window is adjusted by the window control module according to acknowledgment frame received by receiving module and timeout situation which is obtained from clock control module. The transmission control module transmits streaming media data according to the size of window, the clock control module calculates and confirms timeout clock value and handles information of other modules and the receiving acknowledgment module transmits acknowledgment frame to the sender according to

received data frame. The algorithm structure is shown in Fig.1:

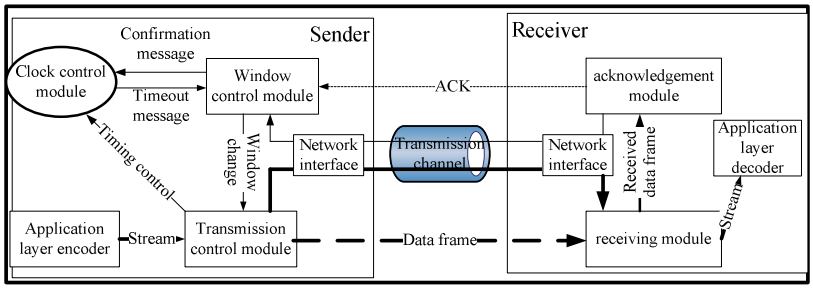

Fig.1 TCP-like Algorithm Structure Model

TCP-like algorithm uses TCP acknowledgment mechanism. RTCP receiving report (SR, RR frame) will be transmited to sender to confirm the received last data frame after the receiving terminal receives RTP data frame required by feedback rate or the clock timeout. The transmission rate of RTP data frame is adjusted by using improved AIMD algorithm control window. In order to achieve more stable rate variation and realize a stronger self-synchronization between flow and control in the stream media transmission based on RTP/RTCP, conversion relationship and control parameters of the original TCP window control mechanism (slow start, congestion avoidance and fast recovery) is improved. At the same time, the retransmission mechanism of TCP is completely abandoned to prevent further congestion and long-term transmission pause caused by waiting.

AVERAGE DELAY JITTER MODEL

The delay jitter of TCP-like algorithm is set as the variation of RTP data frame arrival time. Set Di as the arrival time of i frame and the jitter is expressed by formula (1):

$$Jitter = | D_i - D_{i-1} | \quad (1)$$

During the transmission of TCP-like algorithm, define two adjacent times between package loss with the sign of r≠b, as TD, in which, r is the total number of actually received and lost frames, b is the feedback rate. The window variation of TD period is similar with standard TCP model period of receiving repeated acknowledgment frame. Assuming that Y is correctly received frames during TD period, we can obtain the expected value of Y according to standard TCP model, shown in formula (2):

$$E(Y) = \frac{1-p}{p} + \sqrt{\frac{8}{3bp}} \quad (2)$$

The expected value of W, which is the last window during TD period, is shown in formula (3):

$$E(W) = \sqrt{\frac{8}{3bp}} \quad (3)$$

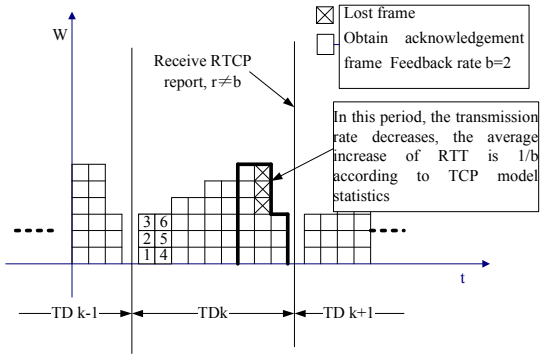

Fig.2 Jitter Variation of TD Period

It is clear that the transmission is in congestion avoidance stage and the feedback rate b is 2 in each TD period, as shown in Fig.2 . It can be conclude that the delay of each frame is RTT/2 and the jitter delay is 0. When the TD period ends, the average delay of received frame will increase if has frame loss, what is similar with standard TCP receiving repeated acknowledgment. Through standard TCP model analysis we can know that the average delay

jitter is RTT/ (2b) at this time. Therefore, the delay jitter of first frame at the beginning of TD can be set as RTT/ (2b) in the time of starting TD period.

Timeout will occur when network has congestion. The experiment shows that the probability of continuous timeout is very low. The paper assumes that, one timeout will occur after n continuous TD periods during the transmission. Set the average value of Timeout is T0 and period of waiting-timeout is Tout. Define the period of including continuous n TD and one Tout as TP. The whole transmission can be deemed that it is composed of a series of continuous TP period. Thus the expected value of delay jitter of frame transmitted in one TP period can be used as the expected value of whole transmission process. The jitter analysis model of TCP-like algorithm in a TP period is shown in Fig.3.

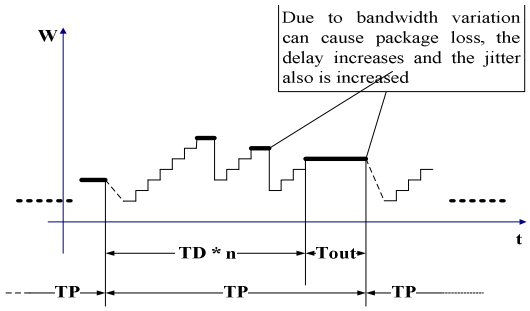

Fig.3 Jitter Analysis Model of TCP-like Algorithm

After the occurrence of timeout, the transmission delay will be T0/2~T0. Assuming that the transmission delay is T0, the delay of last frame of TP period received by receiving terminal will be T0 and the jitter delay sum of this period is |RTT/2-T0|. It can be abstractly thinked that the last frame's delay jitter of TP is |RTT/2-T0|, and the delay jitter of frame transmitted in TD and Tout period is 0. The transmission delay will gradually vary to RTT/2 from T0 in a period of TP starting. It can also be abstractly thinked that the delay jitter of first frame of TP is |T0-RTT/2| and later, the jitter in TD period's steady state is 0 again. In a period of timeout, the specific delay variation is shown in Fig.4.

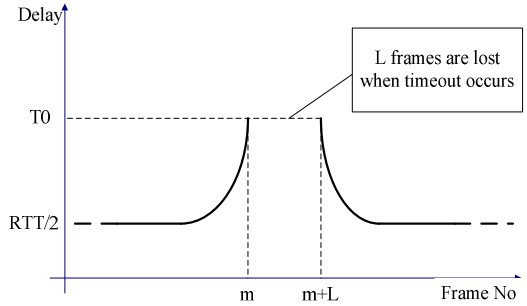

Fig.4 Delay Variation of a Period under the Situation of Timeout

The expected value of average jitter can be obtained based on above model analysis, assuming that the transmitted lost frame is up to E(W) actually in Tout period, shown as formula (4).

$$E(Jitter)=\frac{\frac{1}{E(n)}\mid 2T0-RTT\mid +(1-\frac{1}{E(n)})(\frac{RTT}{b})}{E(Y)-\frac{1}{E(n)}E(W)} \quad (4)$$

In the formula, 1/E(n)can be understood as the timeout probability of TD period, set Q=1/E(n) and (4) can be expressed as (5):

$$E(Jitter)=\frac{Q(2TO-RTT)+(1-Q)(\frac{RTT}{b})}{E(Y)-Q\cdot E(W)} \quad (5)$$

It will be obtained from standard TCP model that, when p->0,

$$E(Q)=\frac{1}{E(W)}=\sqrt{\frac{3bp}{8}} \quad (6)$$

Substitute formula (2) (3) (6) into (5), the estimated value of jitter can be obtained as formula (7) after finishing:

$$E(Jitter) = \frac{p(2T0\sqrt{\frac{3bp}{8}} + \frac{RTT}{b} - \frac{(1+b)RTT}{b}\sqrt{\frac{3bp}{8}})}{1-2p+\sqrt{\frac{8p}{3b}}} \quad (7)$$

When p is small, the denominator of formula (1) is about 1, and then the estimated value can be expressed as formula (8):

$$E(Jitter) = p(2T0\sqrt{\frac{3bp}{8}} + \frac{RTT}{b} - \frac{(1+b)RTT}{b}\sqrt{\frac{3bp}{8}}) \quad (8)$$

PROOF OF THE MAIN RESULT

In order to investigate practical application performance of the algorithm, the paper uses OpenPhone software terminal, which integrates TCP-like algorithm, and implements the experiment under the environment of windows. The experiment environment configuration is shown in Fig.5.

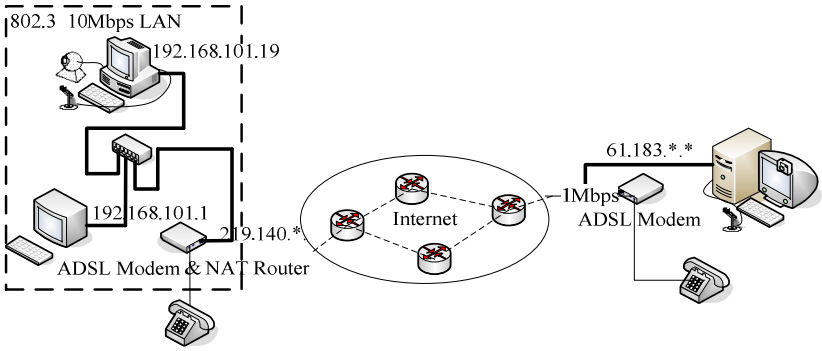

Fig.5: Experiment Environment Configurations

DELAY JITTER

The actually measured average jitter during 1 hour's transmission process, which is calculated in formula (8), is shown in Tab.1. Compared with the actual measurement, the absolute error of the transmission jitter estimated by formula (8) is less than 1ms. This paper is intended to prove that TCP-like algorithm suppresses jitter effectively and provides a more stable transmission rate to improve overall quality of business.

Tab.1: Measurement and Estimation of Delay Jitter

	Average bit rate of interference program (Unit: Kbps)	Package loss rate (Unit: %)	T0 mean value (Unit: s)	RTT mean value (Unit: s)	Jitter measurement value (Unit: s)	Jitter estimation value (Unit: s)
1	175Kbps	0.92%	1	0.374	0.002068	0.002857
2	180Kbps	0.74%	1.022	0.376	0.002986	0.002259
3	185Kbps	1.02%	1	0.373	0.002442	0.003155
4	190Kbps	0.79%	1	0.376	0.002744	0.002382
5	195Kbps	0.93%	1	0.379	0.002864	0.003326

CONCLUSION

The average package loss rate of TCP-like congestion control algorithm is less than 1%, compared with RTP/RTCP algorithm 2%-10%. The proposed algorithm not only reduces package loss rate greatly, but also responses variation of network bandwidth timely. It will provide a more stable transmission rate because the congestion and jitter have been restrained effectively based on the TCP-like algorithm. Moreover, Wireless network will play more and more important role in the future network.The challenges of streaming media business posed by wireless network is inevitable and so the further study will be focused on congestion control strategy for wireless streaming media.

Acknowledgments

The National Natural Science Foundation of China (U1204609); The Education Department of Henan Province Science and Technology Key Project (14A510011); The Youth Science Foundation of Henan Normal University (2012QK21)

REFERENCES

[1] Shiang, Hsien-Po,vanderSchaar, *IEEE Transactions On Multimedia*,Vol.14,No.3,**2012**, 896-909.
[2] Chang Ray-I, Wang Te-Chih,Wang Chia-Hui, *Engineering Applications Of Artificial Intelligence*, Vol.25, No.7, **2012**, 1322-1330.
[3] HayderNatiqJasem,Zuriati Ahmad Zukamain,MohamedOthman, *International Journal of Computer Science and Network Security*, Vol.8, No.10,**2008**, 331-338.
[4] Li Guodong,Jin Pengfei, *Journal of Information and Computational Science*, Vol.9, No.16, **2012**, 4691-4697.
[5] SathiaseelanArjuna, FairhurstGorry, *Computer Communications*, Vol.34, No.15, **2011**, 1836-1847
[6]Attiya, Gamal, *International Journal of Computer Science* Issues, Vol.9, No.2, **2012**, 368-377.
[7] XIAO Fu,WANGRuchuan,SUNLijuan, *Computer Science*, Vol.37, No.7, **2010**, 50-53(in Chinese)
[8]Wan Shanshan, RenZhifeng, *Advances in Intelligent and Soft Computing*, 2012, Vol.133, No.20, **2012**, 371-378.
[9]Shiang Hsien-Po1, Van Der SchaarMihaelal, *IEEE Transactions on Multimedia*, Vol.14, No.3:,**2012**,896-909.
[10] Li Shiyong,Sun Wei, Zhang Yaming, *Journal of Computational Information Systems*, Vol.7, No.14, **2011**, 5292-5299