An Adaptive Algorithm for Real-Time Data Transmission in Multi-Hop Overlay Networks

Ying He, Feng Xiao, Weiwei Fang, Rui Wang, Depei Qian
Sino-German Joint Software Institute
Beihang University, Beijing, P.R. China

yinghe.buaa@gmail.com , xf_friday@yahoo.com.cn , fangvv@gmail.com , rui.wang@jsi.buaa.edu.cn , depeiq@buaa.edu.cn

Abstract

Aiming at improving the performance of real-time data transmission in multi-hop overlay networks, an adaptive flow control transmission algorithm is proposed. The approach considers the reality of low bandwidth and the requirement of high throughput in real-time media streaming so that it takes the forwarding capability of relay peers as the main evaluation metric. Probe data packets are introduced to real-time data transmission to help derive necessary feedback information. Receivers periodically calculate their available buffer size and send feedbacks to senders who use them to adjust the transmission rates. The mechanism for the dynamic adjustment of bandwidth requirements in media streaming is based on an adaptive rate threshold which adjusts the transmission rate according to the available bandwidth. Redundant flow resources are distributed to multiple relay links. Theoretical analysis and simulations indicate that the cooperative transmission algorithm is able to decrease the transmission failure rate and reduce the whole transmission delay of real-time applications in multi-hop overlay networks.

Keywords: Multi-hop Overlay Networks, Cooperative Route, Distributed Algorithm, Peer-to-Peer Media Streaming.

I. INTRODUCTION

The emergence and development of peer-to-peer overlay networks give directions to the network architecture evolution as it can support new applications without upgrading the underlying network infrastructure. Various kinds of high-speed networks have made it feasible to implement real-time media streaming applications such as Skype [1][2] and PPLive[3]. Peers in such networks cooperate with each other and relay streaming data through multi-hop route. Overlay networks improve the current best-effort Internet's incapability of offering any quality of service to the media streaming over the Internet.

Two main categories are the multi-tree-based [4] and mesh-based [5] networks. Multi-tree initialization and construction usually cause a large amount of memory cost and time consumption. Moreover, there is no guarantee for reliable streaming data transmission. The mesh-based networks comparably present better performance in real-time transmission. As real-time streaming has the requirement of high bandwidth, low packet loss and low delay by nature, which may easily lead to network congestion, many techniques and mechanisms are proposed to alleviate network congestion so as to achieve better performance. As each peer is a network endpoint, the bandwidth is always limited especially like a cell phone. Therefore, it seems really important to figure out how to provide high bandwidth and high quality real-time streaming with limited memory. Without effective rate control mechanism, it probably leads to transmission bottleneck such as high packet loss rate and media discontinuity.

Traditional connection-oriented transport protocol – TCP – adds its own flow control and time windowing scheme that may cause media disorder in real-time data transmission. Moreover, TCP is a reliable protocol whose retransmission mechanism would bring further jitter and packet delivery delay. Real-time data transmission is tolerant to packet loss to some extent but not much end-to-end delay. Connectionless transport protocol – UDP – is generally adopted in real-time data transmission. There are two protocols called RTP and RTCP used as accompaniment. However, as an end-to-end protocol, RTP itself does not provide any mechanism to ensure timely delivery or provide other quality-of-service guarantee. It is unaware of any information of receivers and offers no reliability mechanisms. As the bandwidth and computing resource of each endpoint in the network is limited, it is necessary to adjust transmission rate based on the available bandwidth and processing capability of receivers in order to enhance the transmission control efficiency.

Inspired by RTP and RTCP protocol, this paper aims at general mesh-based networks and proposes a real-time data transmission algorithm. With this algorithm, the sender adjusts its sending rate according to the feedback information from the receiver's buffering state. The transmission control mechanism helps to achieve both efficiency and flexibility by avoiding buffer overflow or underflow. In Section II we give a brief introduction of related works in real-time media streaming area. The proposed algorithm for real-time data transmission is

CHINACOM 2010, August 25-27, Beijing, China

DOI 10.4108/chinacom.2010.121

concretely demonstrated in Section III. We analyses algorithm performance and proposes some key points in Section IV. Simulations are designed and implemented in Section V. In Section VI we draw conclusions of this paper.

II. RELATED WORKS

The RTP and RTSP are the two protocols that are currently used for media streaming. RTP accompanied by RTCP is designed for media transport among endpoints. RTSP is used for streaming sessions control by establishing and controlling time-synchronized streams of continuous media. One scalable feedback control mechanism for video sources has been proposed in [6]. The end-to-end mechanism is employed and network states are defined according to the feedback information from the receivers. There is no need for an explicit probing mechanism as the reports from the receivers are multicast periodically with RTP. Feedback controller is introduced in [7] and [8].

With the development of streaming video, congestion control gradually takes the form of rate control [9]. The main purpose of rate control is to minimize the possibility of network congestion by matching the rate of video stream to the available network bandwidth. The so called equation-based control approach is proposed to provide a relatively smooth congestion control for real-time traffic. The TCP-friendly rate control protocol (TFRCP) [10] and the TCP-friendly rate control (TFRC) [11] are two important equation-based rate control protocols. These protocols are based on a throughput model of a TCP connection. The proposed equation helps to determine the sending rate of the video stream that could avoid traffic congestion in a similar way of TCP.

Our algorithm is mainly based on RTP and RTCP with proposed adaptive data transmission approach according to real-time buffering states.

III. PROBLEM ANALYSIS AND ALGORITHM DESCRIPTION

A. Bottleneck of real-time data transmission

Mesh-based networks are generally built as superposition of multiple diverse trees where participating peers employ a swarming content delivery mechanism over a recent window of content [12]. The source media is divided into data slices that are delivered to different levels of multiple trees. Peers in different levels are perceived as relays mutually contributing data slices as media supplement when desired. The purpose of the layered construction and transmission is to offload the media source. As for a network peer, its playback criterion is just in connection with the requirement of the lowest bandwidth. Part of random mesh network for real-time media streaming is shown in Figure 1.

According to the data showed by PPLive, the sending rate of the media source is generally from 381 kbps to 450 kbps, and HD-Video is about 700 kbps and even more. Therefore the bottleneck of real-time data transmission mainly consists in the buffering performance of overlay relay peers. Our paper aims at adjusting the sending rate according to the buffering state of relay peers. With the proposed adaptive traffic control algorithm, the effective transmission rate is improved compared with RTP/RTCP, and in the situation of multiple stream coexistence the loss rate increases not intensely.

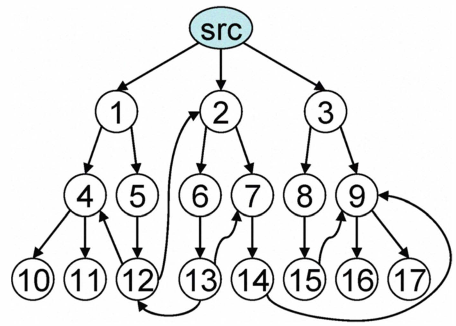

Figure 1. Mesh-based network for real-time media streaming

B. Predefined Conditions

During the data transmission, the sending rate of any peer should not exceed the upper transmission bound of its next relay peer in order to avoid catastrophic failures. However, acquiring real-time transmission capability of the relay peer seems rather difficult. In our algorithm the buffering state is detected periodically. By analyzing the feedback information and adjusting the sending rate, optimized transmission is guaranteed to avoid starvation or media overflow. Several terms used in our algorithm are depicted as follows.

Probe packets – data packets that are transferred between overlay network peers and used to detect the buffering information of receivers. *Delay of the ring path* (equivalent to the round trip time) is calculated according to buffering feedback information. *Rate threshold* – the maximal sending rate that a peer could send to its next relay peer. This value is calculated and adjusted according to real-time buffering information. *Threshold adjustment cycle* – cycle time that the rate threshold is adjusted to accommodate current network state.

C. Algorithm Abstraction

The proposed algorithm mainly contains two phases. When there is idle time which means the network is smooth and with no congestion, the probe packet is sent to detect the buffering state of the receiver. The length of media steaming is calculated according to the available buffer size. Each link of network peers is assigned a lower rate threshold. If the arriving rate exceeds this rate threshold, the media stream will be splitted and distributed to some other nodes. Cooperative transmission needs to split media stream. If different data slices arrive in disorder, the delay is set to the maximal arriving time. If transmission failure happens to any relay peer, the correspondent data has to be retransmitted.

The algorithm of splitting and recombination of media stream is depicted as follows. In order to avoid more delay caused by encoding and decoding in real-time data transmission, our algorithm employs a short buffer split policy.

Suppose the buffer size is L and Link(A,B) needs to split the steam, and then media data is transmitted as the size of rate threshold progressively. In the meanwhile, the deployed buffer size is free for new arriving media. Recombination includes a receiving buffer. Each part of rate threshold data is ready for queuing. Once the queuing delay exceeds the maximal defined delay, the media data is dropped immediately.

D. Algorithm Description and Illumination

Given the source peers set $Src = \{s_1, \mathrm{L}\ s_n\}$, relay peers set $\mathrm{Re} = \{re_1, \mathrm{L}\ , re_m\}$ and end peers set $Dest = \{d_1, \mathrm{L}\ , d_g\}$. Any source peer could also be a relay peer. In the overlay routing table of any given peer s_j, assign each link a rate threshold η_j and an average round trip time art_i. Each peer maintains a routing table lookup time t_{route}. The proposed algorithm process is described as follows.

1) *When a peer receives the first data packet of the relayed media stream, it should immediately look up routing table to search p candidate route entries. The whole routing table lookup time is denoted as t_{route}.*

2) *Send probe packets to the p candidate peers to detect available buffer size. Until the first round trip response packet p_{rob0} returns back, record the average round trip time art_0 and available buffer size λ_0.*

3) *Send new probe packets again after which media stream is immediately transmitted. The transmission rate is adjusted by relevant feedback information.*

The main idea of the algorithm is to define an interval time $\Delta t = t_{route} + art_i$ and media data transfers in a η_i speed. For each average round trip time, one probe packet is sent followed by the media data. The transmission sequence is like $\{p_{rob0}, p_{rob1}, d_{ata1}, d_{ata2}, d_{ata3}, \ldots d_{ataj}, p_{rob2}, \ldots\}$ and the correspondent transmission timing diagram is shown in Figure 2.

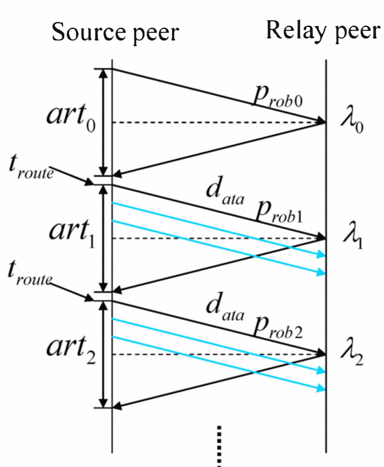

Figure 2. Transmission timing diagram

Our algorithm employs the equation $\Delta t = t_{route} + art_i$ as when the next peer receives media data, it needs to look up the routing table and transfers data. As t_{route} is far less than art_i, t_{route} is neglected for simplification in the process of calculating adaptive transmission rate mentioned below and simulation in Section III. One more condition should be mentioned which is $\eta_0 < \lambda_0 / art_0$ as when the second probe packet does not return back, buffer of the relayed peer overflows. In the process of session initiation, we suppose that λ_0 is a priori for the relay peer. In the real world, the relay peer may need to do some computations to determine this size based on its memory resources and its assessment of network conditions.

Supposing that the two continuous average round trip time are art_i and art_{i-1}. And the corresponding available buffers are detected as λ_i and λ_{i-1}. Difference of buffer size is denoted as $\Delta l_{(i,\,i-1)} = \lambda_i - \lambda_{i-1}$. The media transmission rate

$$v_{cal(i)} = \frac{\Delta l_{(i,\,i-1)}}{\frac{art_i}{2} + \frac{art_{i-1}}{2}} = \frac{2\Delta l_{(i,\,i-1)}}{art_i + art_{i-1}} \quad (i = 2,3,4\ldots) \quad .$$

Considering that the relay peer just received media data and is ready to look up the routing table and send the next probe packet, the actual media input queuing is represented as $\lambda_{act(i)} = \lambda_i - (v_{cal(i)} - \eta_i) * \frac{art_i}{2} \quad (i = 2,3,4\ldots)$. Therefore the actual sending rate should be set to

$$v_{act(i)} = \begin{cases} v_{acl(i)} & (\lambda_{act(i)} > \lambda_i) \\[2ex] \dfrac{\lambda_{act(i)}}{\frac{art_i}{2} + \frac{art_{i-1}}{2}} & (\lambda_{act(i)} \le \lambda_i) \end{cases}$$

According to the above reasoning, when there is redundant buffer, the transmission rate will be increased otherwise will be decreased to avoid overflow. The relevant transmission sequence diagram between peers is shown in Figure 3.

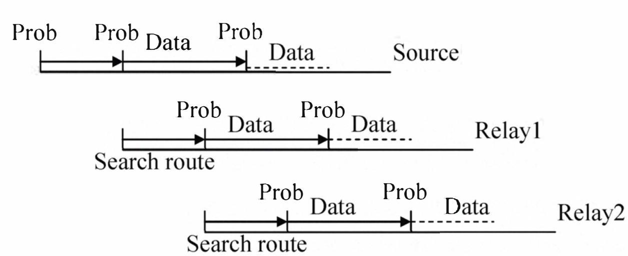

Figure 3. Transmission sequence diagram between peers

IV. ALGORITHM ANALYSIS

In this section, we analyze the algorithm in three main aspects. The first key factor is the buffer utilization rate. In our algorithm, transmission rate of relay peers is calculated according to the buffering state of receivers. On the premise of avoiding buffering underflow and overflow, the maximum available buffer is used to calculate the real acceptable transmission rate. With the buffering state changes, the transmission rate is almost optimized. Therefore the second factor – total time lag – is relevantly decreased.

In the circumstance of multi-hop media stream coexistence, effective transmission rate is another important issue. Denote r as the media streaming paths and each path has several end to end links. R is the set of all paths in the overlay networks that is $r \in R$. Define f as media stream and $y_r^f(t)$ as the throughput that f transmits along path r. Correspondingly, the failed media streaming is denoted as $\upsilon_r^f(t)$. The average effective transmission rate could be expressed as

$$\rho(G) = \frac{\sum_{f\in F}\sum_{r\in R_f}\int y_r^f(t)}{\sum_{f\in F}\sum_{r\in R_f}\int y_r^f(t) + \sum_{f\in F}\sum_{r\in R_f}\int \upsilon_r^f(t)}.$$

V. SIMULATIONS

We have implemented the adaptive rate control transmission algorithm in NS2 [13] simulator where the RTP/RTCP module is employed and compared. The relevant simulation topologies are shown in Figure 4 and Figure 5. The bandwidths between routers and peers are 10Mbps, and the delays are 6ms. The bandwidth between routers is 10Mbps and the delay is 10ms. Access link bandwidth is provisioned sufficiently so that congestion can only occur in the bottleneck links. For ease of simulation description, the proposed algorithm is named as ABUF.

In the first experiment, we adopt the one bottleneck simulation topology. Media stream transmission path is selected as $S_1 \rightarrow R_1 \rightarrow S_2 \rightarrow D_1$. The packet size is 512 Bytes and the initial transmission rate is 500 kb/s. The rate threshold is 125 kb/s. In Figure 6 we can see that the loss rate of ABUF is lower than RTCP since ABUF adjusts its transmission rate according to the feedback information of relay peers while RTCP does not maintain such information. Figure 7 shows the comparison of effective throughput. For RTCP there is nearly a linear-prone state as it has no record of buffer information. ABUF presents a waved-prone state which demonstrates its buffer jitter state. If there is available buffer size for relay peers, the sending rate of source peer will give it a best-effort transmission rate according to the proposed algorithm. When there is a buffer overflow detected, the sending rate will decrease. This mechanism ensures that more effective media stream is received.

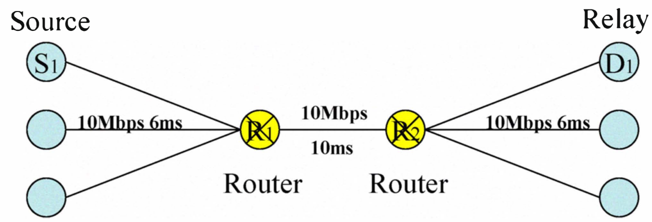

Figure 4. One bottleneck simulation topology

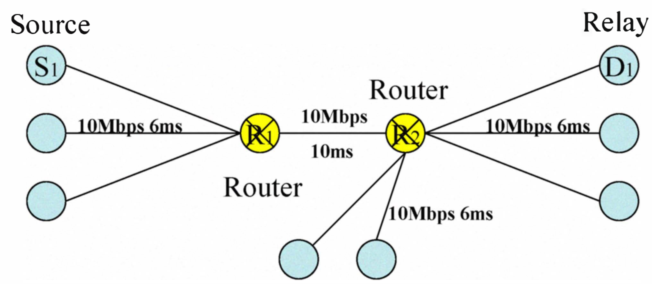

Figure 5. Two bottleneck links simulation topology

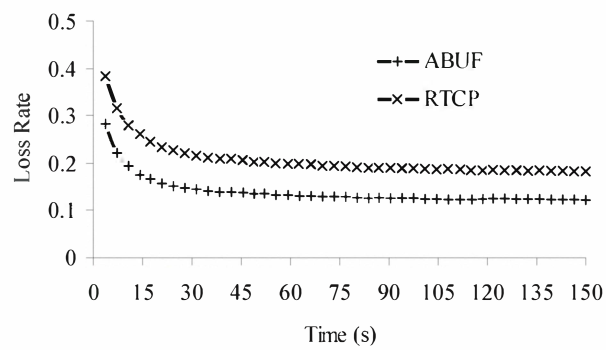

Figure 6. Loss rate comparison with transmission for one bottleneck link

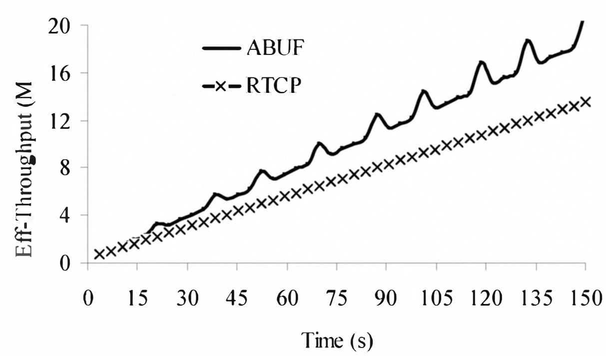

Figure 7. Effective throughput state comparison with transmission for one bottleneck link

The second experiment topology includes 100 peers and 700 links where each two peers maintain a data stream. Figure 5 shows a two bottleneck links topology and we also carry four and six bottleneck links for two peers' transmission. The average effective transmission rate of the network is calculated as $\rho(G)$ mentioned in Section IV. We get the average loss rate in Figure 8. We can see that for multiple stream coexistence, the loss rate will increase while the difference is relevantly low. With the transmission going on, a comparatively steady transmission state is maintained.

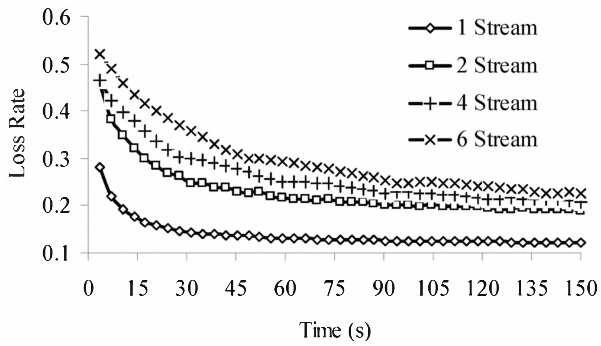

Figure 8. The average loss rate for different number of streams between two peers in ABUF

VI. Conclusion

In this paper, we propose an adaptive algorithm for real-time data transmission in multi-hop networks. The algorithm adds probe packets for detecting the buffering state of relay peers. According to the feedback information, the source peer could calculate the available bandwidth accurately and control an optimizing transmission rate that helps to avoid underflow and overflow. Theory analysis and simulation indicate that the proposed algorithm could decrease the stream loss rate and increase effective throughput through adaptive transmission control. In the situation of multiple stream coexistence, the average effective transmission rate does not decrease significantly, which ensures the real-time transmission quality in multi-hop overlay networks.

Acknowledgment

This work was supported by the Natural Science Foundation of China under the contract 60673180 and National High Technology Development Program (863 Program) under Grant No. 2009AA01Z144.

References

[1] S. A. Baset and H. G. Schulzrinne. An analysis of the skype peer-to-peer internet telephony protocol. In Proc. Of IEEE Infocom, 2006.

[2] Dario Bonfiglio, Marco Mellia, Michela Meo, Dario Rossi, Paolo Tofanelli. Revealing skype traffic: when randomness plays with you. Sigcomm, 2007.

[3] Yan Huang, Tom Z. J. Fu, Dah-Ming Chiu, John C.S. Lui and Cheng Huang. Challenges, Design and Analysis of a Large-scale P2P-VoD System in SIGCOMM 2008.

[4] M. Castro, P. Druschel, A.-M. Kermarrec, A. R. A. Nandi, and A. Singh, "SplitStream: High-bandwidth content distribution in a cooperative environment," in ACM SOSP, 2003.

[5] Nazanin Magharei, Reza Rejaie, Yang Guo. On reducing Mesh Delay for Peer-to-Peer Live Streaming, in INFOCOM 2008.

[6] J.-C. Bolot, T. Turletti, and I. Wakeman, "Scalable feedback control for multicast video distribution in the internet," in *SIGCOMM Symposium on Communications Architectures and Protocols*, (London, England), pp. 58-67, ACM, Aug. 1994.

[7] H. Kanakia, P. Mishra, and A. Reibman, "An adaptive congestion control scheme for real-time packet video transport," in *SIGCOMM Symposium on Communications Architectures and Protocols*, (San Francisco, California), pp. 20-31, ACM/IEEE, Sept. 1993.

[8] I. Busse, D. Cochrane, B. Deffner, P. Demestichas, B. Evans, W. Grupp, N. Karatzas, K. Kassapakis, P. Legand, Z. Lioupas, Y. Manolessos, K. Nagel, T. Pecquet, B. Rabere, G. Seehase, G. Stamoulis, M. Scholz, H. Schulzrinne, and S. Vaisenberg, "Quality module specification (D7)," Deliverable R2116/ISR/WP2/DS/S/007/b1, RACE Project 2116 (TOMQAT), Dec.1994.

[9] D. Wu, Y. T. Hou, and Y.-Q. Zhang, "Transporting real-time video over the Internet: Challenges and approaches," *Proc. IEEE*, vol. 88, pp. 1855-1875, Dec.2000.

[10] J. Padhye, J. Kurose, D. Towsley, R. Koodli, A model based TCP-friendly rate control protocol, in: NOSSDAV, NJ, USA, 1999.

[11] M. Handley, S. Floyd, J. Padhye, RFC 3448 – TCP friendly rate control (TFRC): protocol specification, January, 2003.

[12] AD. A. Tran, K. A. Hua, and T. T. Do, "A Peer-to-Peer Architecture for Media Streaming," in IEEE JSAC: Special Issue on Advances in Overlay Networks, Vol.22, N.1, Jan. 2004.

[13] http://www.isi.edu/nsnam/ns/