A Bandwidth Allocation Algorithm Based on Historical QoS Metric for Adaptive Video Streaming

Ling Guo[1], YuanChun Shi[1], and Wei Duan[2]

[1] Key Laboratory of Pervasive Computing, Tsinghua University
{Guoling02@mails.,shiyc@}tsinghua.edu.cn
[2] China United Telecommunications Corporation
duanw@chinaunicom.com.cn

Abstract. This paper introduces a dynamic bandwidth allocation algorithm in a video streaming multicast system. The approach is to introduce the vibration of received video quality into the QoS metric and make the receivers more negative in subscribing higher layers when bandwidth increases. A simulated annealing algorithm is applied in the server side to find the optimal allocation schema within the concurrent network situation at run time. Simulated experiments on NS-2 have been carried out to validate the algorithm. The result shows an improvement of 6.8 percents increase in received data rate and 6.0 percents decrease in data loss rate.

1 Introduction

The Internet has been experiencing explosive growth of audio and video streaming. Researchers have developed layered video multicast to provide video streaming to a large group of users. Various devices such as PDA, desktop, laptop, even mobile phones are widely used in various network conditions, as diversely as network conditions, such as LAN,ADSL,GPRS and etc. Layered video codec is used to suit the heterogeneous environment[1].

As we have mentioned above, the perceptual quality of a video is determined by many factors, such as image size and frame rate. Besides, Internet applications desire asymptotically stable flow controls that deliver packets to end users without much oscillation[2]. In a best-effort network, most of the video streaming systems use flow control mechanisms like AIMD to be fair to other applications. AIMD is known for drastically decrease of accept window when timeout or data loss occurs. The oscillation of bandwidth makes it even more difficult to get a stable video streaming over Internet. Recently, many approaches of congestion control have been raised to avoid fluctuation in video quality [2, 3]. Some others devised reschedule mechanisms of the buffered data to compensate the network delay or jitter [11]. In a layered video streaming system, comparing to network congestions, the bandwidth allocation mechanism have greater impact on the traffic. As far as we know, no work has been ever done in exploring the feasibility of improving the allocating mechanism to gain more stable video streaming.

P. Lorenz and P. Dini (Eds.): ICN 2005, LNCS 3420, pp. 917–925, 2005.

The rest of this paper is organizing as the following. Part two introduces related works. The third part discusses the metric of continuous QoS. Then, the bandwidth allocation algorithm which uses simulated annealing is presented in part four. Then part five gives out the experimental result on NS-2 simulation to validate the allocation algorithm. In Part six, the paper ends with future work and conclusion.

2 Related Works

To transmit video packets over Internet, researchers have extensively explored many possibilities.

At first, sender-driven-congestion-control for adaptively encoded video was proposed and developed in unicast filed. The key point of the method is to adjust its encoding data rate in accordance with the network condition. [5,6,7]. The sender-driven algorithms are also extended to multicast, but as the video has only one layer, if the group has a low bandwidth node the whole multicast group will be impacted.

Receiver-driven adaptation algorithms were proposed after the emergence of layered video. The video source generates a fixed number of layers, and each user attempts to subscribe as many layers as possible. With the development of layered codec, it is possible to dynamically adjust the amount of layers as well as the data rate of each layer. Algorithms that take advantage of this improvement came into scene, such as SAMM (Source-Adaptive Multilayered Multicast Algorithms) [8]. Some of the layered algorithms set priority on layers and expect the network nodes selectively drop the higher layers when congestion occurs. Some other approaches just admit the concurrent infrastructural Internet as a QoS unaware network and try to compensate it in the application level. One of them is proposed by Liu [1, 4]. The paper describes a method to find the optimal allocation schema by a recursive function within an acceptable overhead. But the algorithm doesn't consider bandwidth vibration. It always try to make full use of the bandwidth.

As to perceptive QoS, many other aspects left unconsidered in the most QoS metrics, such as intra-frame synchronization and constant quality of video streams. Reza [9] managed to reveal the important impact of buffer and congestion on the perceptual QoS of video streaming. Reza points out that in order to gain smooth video, the buffer should always have enough data and can survive at least a TCP back-off in the near future. Therefore, when bandwidth increases, instead of simply joining a higher layer, the author proposes that the allocation algorithm should make sure that buffers should always have enough data to survive at least a TCP back-off. They also propose a method to allocate bandwidth among the active layers to prevent buffer underflow.

As it discusses above, the bandwidth allocation algorithm is designed to make full use of the available bandwidth, but they usually failed to consider some temporal requirements that intrinsically lay in streaming video. While researchers in congestion control reveal to us the relationship between jitter and bandwidth utilization, but the method of congestion control is not direct and may cause some side effects. Based on this, we propose a bandwidth allocation algorithm that integrates temporal characters.

3 Historical QoS Metric

3.1 QoS of Streaming Video

The quantitative metric of QoS is the basic of the allocation algorithm. In a dynamic environment such as the Internet, both user and service-provider factors are variable. The end-user wants to make full use of the bandwidth while the service provider pursues cost-effectiveness of bandwidth resources and the end-systems'QoS. It is a trade-off to decide which one to use. Some peer-to-peer systems use received data as QoS metrics. Nevertheless, multicast applications usually take the overall cost-effectiveness as the metrics. Usually, the computation resources and the output bandwidth of the server are limited. The server should be fair and efficient in allocation resources. One example of the cost-effective metric is the bandwidth utility [1,4]. In this paper, bandwidth utility is used as the basic QoS metric.

$$q = \frac{r}{b} \tag{1}$$

where r is the received data rate and b is the available bandwidth of the receiver.

3.2 Continuous Video Quality

Usually in video streaming systems, the end-user has a data consumption rate. The consumption rate is decided by the decoder and not necessarily constant. When the received data rate is lower than the consumption rate, a jitter will take place. In the best-effort Internet, the vibration on bandwidth happens constantly.

The continuality of video streaming also has much to do with history, which refers to the QoS performance in the past. Apparently, if the video quality increases drastically and decrease abruptly, it will cause discomfort change to users. In a multicast video streaming system, changes in QoS are mainly caused by the change of video layers. To introduce a history factor into QoS metric in such a system is our attempt.

3.3 Streaming System Model

Firstly, a system model is introduced as the basis for further discussion. It is real-time video streaming system using a TCP friendly application level multicast protocol. The server uses a layered codec to produce M layers. The cumulative data rate vector of M layers is $\rho^{'} = \{p_0, p_1, ...p_M\}$. There are N receivers $\{R_i \mid 0 \le i < N\}$ connected in through heterogeneous networks. At time t, receiver R_i subscribes $c_i(t)$ layers and has a received data rate $\omega_i(t)$. Totally, a source data flow with the rate of $p_{c_i(t)+1}$ would be sending to user i at time t. In addition, in every time span Δt, each receiver R_i measures its own bandwidth $\Gamma_i(t)$ and reports it to the server.

Meanwhile, the server detect its bandwidth capability B(t) every Δt. Based on these reports, the server adjusts the allocation schema to get an optimal overall QoS:

$$M(t,L)=\sum_{i=0}^{n-1}\mathrm{Q}_i(t,l_i) \tag{2}$$

where $L=\{l_0,l_1...l_i...l_{n-1}\}$ is the vector of the new allocation schema and $\mathrm{Q}_i(t,l_i)$ is the estimated QoS of receiver R_{i} with l_i layers. The QoS metric will be discussed below. After the server finds out an optimal allocation schema, it will send notifications to receivers who need to drop or join a layer.

3.4 Bandwidth Burst

Consider the condition when a bandwidth burst happens, according to best-effort allocation algorithms, the user will subscribe a high layer immediately. After a while, the bandwidth drops to its average level then the user has to drop the highest one or two layers. This short-term subscribe-drop pair not only brings fluctuation in receiver's QoS but also intrigues buffer underflow and overflow at the receiver's side. What's more, during this subscription and drop process, the server sends out more data than what the receiver can receive. So sometimes when bandwidth bursts occurs, the best-effort bandwidth allocation algorithms will cause a short time of high QoS video and latter a jitter in client's side, we call this saw tooth in QoS, which is not desired by the receivers.

3.5 Historical QoS Metric

We introduce a historical factor into the QoS metric to avoid saw tooth. Suppose $q_i(t)$ is the QoS value of user i. We use bandwidth utility as the QoS metric as in (1). The historical QoS metric we defined is composed of a basic QoS metric and a historical effect factor:

$$Q_i(t,l_i)=q_i(t,l_i)*\chi(\eta_{\mathrm{i}}(t)) \tag{3}$$

Where $\eta_{\mathrm{i}}(t)=\dfrac{\Gamma_i(t)-\Gamma_i(t-1)}{\Gamma_i(t-1)}$ and $q_i(t,l_i)$ is the basic metric, $\eta_{\mathrm{i}}(t)$ is the bandwidth change variation. $\chi(\kappa)$ is the effect function of $\eta_{\mathrm{i}}(t)$. The goal of this function is to reduce the possibility of subscribing higher-level layer when the bandwidth increases. When $\eta_{\mathrm{i}}(t)>1$, the history effect factor of $\chi(\kappa)$ should be less than one. The higher $\eta_{\mathrm{i}}(t)$ is, the less the effect factor is.

Now with the new QoS metric, R_{i} has a much lower estimated QoS value when bandwidth bursts. The historical factor is the changing rate of the bandwidth.The more the bandwidth increases, the little the historical factor is. When bandwidth burst happens, $\eta_{\mathrm{i}}(t)$ in (3) is smaller than 1. It is more likely that the bandwidth allocation

algorithm will choose not to add a layer. If in the next time span Δt, bandwidth drops, the receiver will not change the subscription layer. If bandwidth does not drop, it is likely that it is not a burst. Then in the next Δt, the historical effect factor would be one and likely get the opportunity to add a layer. Therefore, and the historical factor makes the allocation algorithm more stable to avoid some short-term subscribe-drop pairs.

As mentioned above, the effect function $\chi(\kappa)$ in (3) should increase slowly when $\kappa > 1$ and remain "1" when $\kappa < 1$. We found $\chi(\kappa) = \begin{cases} e^{-a(\kappa-1)} (\kappa > 1) \\ 1 (\kappa < 1) \end{cases}$ is a simple function fulfilling the requirements:

Fig. 1. Effect function of Vibration on QoS

According to (1) and (3), the quantitative QoS metric of user i at time t is

$$\begin{cases} Q_i(t) = \dfrac{r_i(t)}{b_i(t)} \left(\dfrac{\Gamma_i(t)}{\Gamma_i(t-1)} \le 1 \right) \\ Q_i(t) = \dfrac{r_i(t)}{b_i(t)} * e^{-a * \frac{\Gamma_i(t) - \Gamma_i(t-1)}{\Gamma_i(t-1)}} \left(\dfrac{\Gamma_i(t)}{\Gamma_i(t-1)} > 1 \right) \end{cases} \tag{4}$$

It means that when bandwidth does not increase, the QoS equals the bandwidth utility; if the bandwidth increases, the historical factor is less than 1 and the QoS is less than the bandwidth utility.

4 Dynamic Allocation Algorithm

In dynamic allocation algorithm, we use the QoS metric in (4) to measure the QoS. The problem is to find out a subscription schema to get the maximal overall QoS. For that purpose, a simulated annealing algorithm is used to search for optimal allocation schema. According to (4), the optimization goal of the simulated anneal algorithm is:

$$\sum_{i \in \{\frac{\Gamma_i(t)}{\Gamma_i(t-1)} \le 1\}} \frac{r_i(t)}{b_i(t)} + \sum_{j \in \{\frac{\Gamma_i(t)}{\Gamma_i(t-1)} > 1\}} \frac{r_j(t)}{b_j(t)} * e^{-\frac{\Gamma_i(t) - \Gamma_i(t-1)}{\Gamma_i(t-1)}} \tag{5}$$

The searching space S is the entire possible subscription schema in the current network condition:

$$S = \{C_t(c_0(t)...c_i(t)...c_{n-1}(t)) \mid \sum_{i=1}^{n} p_{c_i(t)} \le B(t)\} \tag{6}$$

While, in this algorithm, we have a constraint on the server side bandwidth:

$$\sum_{i=0}^{n-1} r_i(t) \le B(t) \tag{7}$$

The method to find the next point is important to the efficiency of the algorithm. It can choose a new point as well as examined points. In the experiment, we found that if the possibility of choosing a new point is equal to the possibility of choosing an old one, the algorithm would spent a lot of time hovering between several points and it needs a large MARKOV value to get the optimal point. Therefore, we set different possibility value at new points and old points, the algorithm can get to the optimal very fast.

5 Simulation on NS-2

NS-2 simulation is carried out to validate the algorithm. In the experiment, all the links are duplex links with the delay of 2ms. Queues with FIFO drop-tail and maximum delay of 0.5 sec are used. The maximal package size is set to 1000 bytes. To simulate the layered video streaming, we use a video trace file of a temporal scalable video with three layers. The data rates of three layers are [210.78, 110.92, 60.575kbps]. The transport protocol is UDP.

The simulation topology is like the figure 2 as below. In which, node2/4/5/6/7 are all receivers of the layered video streaming. A FTP flow is set between node0 and node3. The allocation algorithm executes on the server side every 4 seconds. All the simulations run for 500 seconds to get stable results.

The simulated annealing algorithm is used to search for an optimal allocation schema at run-time. In order to demonstrate the effect of the historical QoS factor, we carry out two comparative experiments, A and B. They are all the same except that in A, a historical effect function is used.

In order to get the available bandwidth, we use Packet Pair algorithm [10, 12].

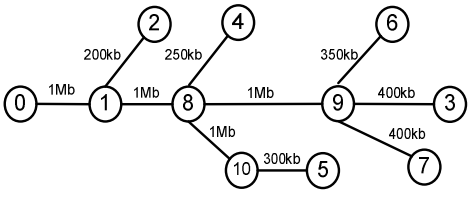


Fig. 2. Topology of the simulation scenario

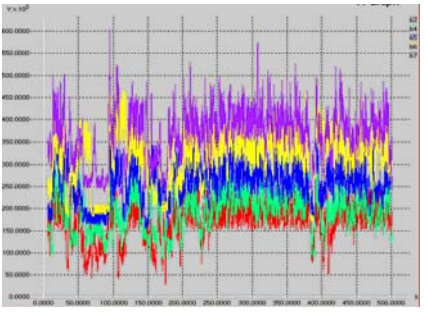
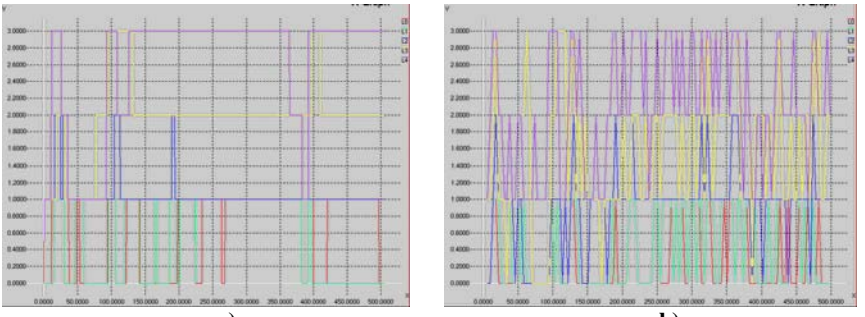


Fig. 3. Available bandwidth of experiment A



a) b)

Fig. 4. Subscription Record of Experiment A and B

In Fig. 4, B has much denser fluctuations than A has. That is because in A, the bandwidth incensement means lower QoS than B does. The historical effect factor is usually lower than "1" when bandwidth increases. Therefore, the possibility for the allocation algorithm to add the layers is lower than B. Fig. 3 is the estimated available bandwidth in experiment A. In Fig. 4, the bandwidth reaches a stable status after a period of adjustment. In Fig. 4 A), subscribed layers increases and decreases as the bandwidth does. For example, in time 0~50, the bandwidth increases and hold for a while. In 30s, bandwidth decreases. Correspondingly, in Fig. 4 A) all the four receivers add a layer in the time span. Then decrease after 30s.

From the statistics in Table 1, A have higher average successfully received data rates than B. In addition, generally data loss ratio is lower in A than in B. Except that in B, Node2 has a lower loss ratio and a slightly higher data rate. Node2 is connected with Node1 through a connection of 200bps, which is lower than the data rate of the first layer. Therefore, the feasible choice of Node2 is to subscribe the first layer or to subscribe nothing. The effect of the historical factor is to reduce the possibility of adding a layer when the bandwidth increases. Moreover, for Node2, sometimes, the bandwidth utility is zero and the historical factor multiple does not have any influence to it.

Table 1. Statistics of experiment result A and B. The shadowed column is the statistic of A and the other is B's. Sending rate is calculated in the server side according to the subscribed layers. Data Rate is calculated according to the successfully received data packet in each node

Node	Bandwidth(kbps)	Sending Rate(kbps)		Loss Rate(%)		Data Rate(kbps)	
N2	200	143.748	145.408	12.508	9.2	125.779	132.030
N4	250	209.043	202.402	3.252	3.583	202.245	195.150
N5	300	275.264	242.915	6.287	8.349	257.958	222.634
N6	350	342.639	308.413	5.139	6.57	325.031	288.150
N7	400	388.490	365.277	3.998	4.473	372.958	348.938

In above, the result shows that historical effect factor improves the video streaming by increasing the data rate by 6.89 percents and decreasing the loss rate by 6.07 percents.

6 Future Work and Conclusion

In this paper, we introduce a historical factor into QoS Metic to get a smoother video. The allocation algorithm is more conservative and helps the multicast video streaming system to maximize the overall QoS through the optimizing of subscribing schema. Simulated annealing algorithm is used to get an optimal allocation schema at run-time. Experiments on NS-2 are conducted to demonstrate the algorithm. Experiment results show that in most cases, the historical effect factor can avoid frequent fluctuation of subscribed layers and improve video streaming QoS.

Further work would include a real implementation with layered codec and heterogeneous network condition. Besides, mobile network connections are not stable and the capability of the mobile device varies. Layered video streaming on mobile network is also widely discussed. The algorithm would be extended to mobile network scenarios.

References

[1] J. Liu, B. Li, and Y.-Q. Zhang, Adaptive Video Multicast over the Internet, IEEE Multimedia, Vol. 10, No. 1, pp. 22-31, January/February 2003.

[2] Min Dai, Dmitri Loguinov, “Analysis of Rate Distortion Functions and Congestion Control in Scalable Internet Video Streaming”, Proceedings of the 13th international workshop on Network and operating systems support for digital audio and video, June 2003

[3] R. Johari and D. Tan, End-to-End Congestion Control for the Internet: Delays and Stability," IEEE/ACM Transactions on Networking, vol. 9, no. 6, December 2001.

[4] Jiangchuan Liu, Bo Li and Ya-Qin Zhang, ”An End-to-End Adaptation Protocol for Layered Video Multicast Using Optimal Rate Allocation”, in IEEE Transaction on Multimedia Vol 6, No. 1, February 2004.

[5] M. Gilge and R. Gusella, “Motion video coding for packet-switching networks——an integrated approach”, in SPIE Conf. Visual Communications and Image Processing, Nov. 1991.

[6] Y. Omori, T. Suda, G. Lin, and Y. Kosugi, “Feedback-based congestion control for VBR video in ATM networks”, in Proc. 6th Int. Workshop Packet Video, 1994.

[7] C.M. Sharon, M.Devetsikiotis, I. Lambadaris and A.R. Kaye, “Rate control of VBR H.261 video on frame relay networks”, in Proc. Int. Conf. Communications(ICC), 1995.

[8] Brett J. Vickers, Cello Albuquerque, and Tatsuya Suda, “Source-Adaptive Multilayered Multicast Algorithms for Real-Time Video Distribution” in IEEE/ACM Transaction on Networking, Vol, 8, NO. 6, Dec. 2000.

[9] Reza Rejaie, and Mark Handley, “Quality adaptation for congestion controlled video playback over the Internet”, in Proc. IEEE Infocom, March 1999.

[10] Gili Manzanaro J., Janez Escalada, L., Hernandez Lioreda, M., Szymanski,M. "Subjective image quality assessment and prediction in digital video communications. COST 212 HUFIS Report, 1991.

[11] Laoutaris N., Stavrakakis I, "Intrastream Synchronization for Continuous Media Streams: A Survey of Playout Schedulers", IEEE Transaction on Network, May-June 2002, Vol, 16, Issue, 3, Pages, 30 - 40

[12] Arnaud Legout and Ernst W. Biersack. PLM: Fast Convergence for Cumulative Layered Multicast Transmission Schemes. In Proc. of ACM SIGMETRICS'2000, pages 13--22, Santa Clara, CA, USA, June 2000.