Improving Media Sensitivity of TCP-Friendly Rate Control for Multimedia Streaming Applications

Adrian Sterca
Babes-Bolyai University
M. Kogalniceanu, No. 1
Cluj-Napoca, Romania
forest@cs.ubbcluj.ro

ABSTRACT

In this paper, we present a TCP-friendly congestion control that is at the same time media-friendly (i.e. sensitive to media characteristics). These types of rate controls are more suitable for multimedia streaming applications than the classical TCP congestion control and smooth TCP-friendly rate controls. We build our media-friendly rate control by combining the notion of TCP-friendliness with a general optimization framework for bandwidth sharing in computer networks. The resulting control is sensitive to network conditions (i.e. TCP-friendly) and also to multimedia stream characteristics (i.e. media-friendly) which makes it more suitable for controlling the transmission rate of multimedia streaming applications and should improve the perceived quality of video streams in best-effort network conditions.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General—*Data communications*; C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Network communications*

Keywords

TCP-Friendly congestion control, multimedia streaming, optimized congestion control

1. INTRODUCTION

In the last decade, multimedia communication and real-time streaming over the Internet have received much attention from the scientific community. Due to their novelty and powerful social impact, multimedia streaming applications like VoD (Video on Demand), video conferencing and live broadcasting are becoming increasingly popular nowadays. However, the heterogeneity and the best-effort nature of the Internet pose great challenges to multimedia streaming applications. This is because no QoS guarantees can be established on the Internet. Due to the time constraints related to multimedia data, multimedia streaming applications need an almost isochronous communication and a guaranteed high bandwidth over time and, ideally, they should never suffer from lacking bandwidth. But these can not be achieved due to the best-effort nature of the Internet, so streaming servers must continuously adapt to changing network conditions. In other words, they must perform congestion control [15].

The first choice for performing congestion control is to use TCP as transport protocol, but TCP's Additive Increase Multiplicative Decrease is not well suited for multimedia streaming applications. TCP's AIMD congestion control treats flows having the same round-trip time and sharing the same bottleneck link identically because it aims at max-min fairness. TCP does not distinct between elastic applications (i.e. applications which can tolerate bandwidth fluctuations, e.g. file transfer applications) and inelastic applications (i.e. applications having strict bandwidth requirements because of real-time constraints, e.g. multimedia streaming applications). There are several characteristics of TCP that makes it rather unsuitable for multimedia streaming applications. First of all, by implementing congestion control and guaranteed retransmission, TCP trades timeliness over reliability: it is more important the data arrives safely and in-order than it is to arrive in time (i.e., bandwidth is sacrificed for retransmissions). This philosophy is counterproductive for multimedia streams, for which timeliness is more important than reliability. Secondly, TCP's congestion control algorithm determines a steep variation of the transmission rate, a variation that is not well coped with by current codecs. Steep degradations of the transmission rate of a multimedia stream has very bad consequences on the quality perceived by the final receiver.

Due to TCP's unsuitability for streaming applications, other TCP-friendly [15] congestion controls were developed which achieve a smoother throughput than TCP's AIMD, the most well-known being TFRC (TCP-friendly Rate Control) [1, 2]. These smooth congestion controls have a more stable throughput than TCP's AIMD because they are less aggressive than TCP in using new available bandwidth, but they are also slower responsive to congestion than TCP. Because they offer a more stable throughput, multimedia streams, especially CBR (*Constant Bit Rate*) ones, but also VBR (*Variable Bit Rate*) ones, can be better adapted to predictable bandwidths by the streaming servers. However, although smooth congestion controls improve the delivery of multimedia streams, they are not the optimal solution, because they don't take into consideration media character-

MoMM 2008, November 24-26, 2008, Linz, Austria.
Copyright 2008 ACM 978-1-60558-269-6/08/0011 $5.00.

istics of the stream (i.e. they are not media-friendly). Sometimes, especially when the client prefetch buffer is small, it is better if the rate control algorithm would consider bitrate and other characteristics of the stream like quality, scene bitrate, etc. in computing the transmission rate.

The rest of the paper is organized as follows. In section 2 we review previous work on which our paper is built. Then section 3 describes the new TCP-friendly and media-friendly congestion control optimized for video streaming applications. The paper ends with section 4 which contains future work of the author.

2. BACKGROUND

The TCP-Friendly Rate Control (TFRC) [1, 2] is a rate-based congestion control that has two main components: the throughput function and the WALI (i.e., Weighted Average Loss Intervals) mechanism for computing the loss rate. The throughput function is the throughput equation of a TCP-Reno source [3]:

$$X_{tfrc}(p) = \frac{s}{R\sqrt{\frac{2p}{3}} + t_{RTO}(3\sqrt{\frac{3p}{8}})p(1+32p^2)}, \qquad (1)$$

where X_{tfrc} is the transmission rate in bytes/sec, s is the packet size, R is the round-trip time (RTT), p is the steady-state loss event rate and $t_{RTO} = 4 * R$ is the TCP retransmit timeout value. This throughput function is behind TCP-friendliness of TFRC. WALI, the mechanism for computing the loss rate as a weighted average of the last 8 loss intervals, is responsible for the smoothness of throughput. Network simulations [2] reveal that indeed TFRC's throughput is more stable than the throughput of TCP.

A different and more general approach in congestion control derived from optimization theory is taken in [4]. Kelly et al. derived two gradient-like algorithms, a primal algorithm and a dual, to control the congestion in the Internet in an end-to-end way and to obtain optimal bandwidth allocation among competing sources. A primal congestion control algorithm is an algorithm which uses a dynamic law (i.e. differential equation) to control the throughput at the source and a static function for computing the price of using the network at the links. A dual algorithm does the opposite: it uses a dynamic function at links (routers) and a static one for source's throughput. Both congestion control algorithms achieve proportional fairness in sharing bandwidth between competing sources. Authors of [4] also proved that both algorithms are stable in the absence of delays. Following Kelly et al.'s paper, several other authors devised different variants of the primal congestion control algorithm and prove their stability properties in the presence of delays [5, 6, 7, 9, 10]. See [14] for an excellent survey. However, all aforementioned papers use general utility functions and they don't consider the specific characteristics of multimedia streaming applications when computing the send rate.

Our work is largely based on the work in [11]. In [11] Low et. al take a dual approach in solving the optimal bandwidth sharing problem. They first use a gradient-descend algorithm at the links for computing the equilibrium congestion prices which are also Lagrange multipliers and then use a simple equation to compute the optimal transmission rates which generate these equilibrium congestion prices at the links. The authors prove that their dual algorithm converges to an optimal bandwidth allocation. However, this dual algorithm achieves a variable fairness between competing sources [12] depending on sources' choice of the utility function and also this congestion control algorithm can be media-friendly (for appropriate choice of utility function) and can achieve TCP fairness (i.e. can be TCP-friendly), but not both at the same time which is exactly what we are looking for.

The work presented in [13] is the closest to our work. Authors develop a media- and TCP-friendly congestion control based on TFRC using a two-timescale approach: they compute the long term average of throughput according to TFRC, but they modify this throughput on a smaller timescale according to the rate of increase/decrease of an utility function obtained from the rate-distortion characteristics of the stream. We have two observations related to this paper. First, the utility function it is used was developed for MPEG FGS (Fine Granularity Scalable) [8] video streams, so the algorithm does not seem to work for other type of streams, while our algorithm works for general, DCT-based, layered scalable video streams. Second, since the rate-distortion utility function is not scaled with TFRC's throughput we are not sure that the derivative of the utility function will have significant influence on TFRC's throughput (even on small timescales) in all network scenarios.

3. A MEDIA-FRIENDLY TFRC

A congestion control for multimedia streaming applications must be fair to TCP (i.e. it must be TCP-friendly), because TCP is the dominant protocol in the Internet. Also such a control must be sensitive to media characteristics when establishing the available bandwidth in order to maximize the perceived quality of the stream at the client side. In the next 3 subsections we start by developing a media-friendly congestion control that optimally shares the bandwidth in best-effort networks and then based on this media-friendly control we build a new congestion control that is media-friendly and TCP-friendly at the same time.

3.1 Optimization framework

We are using the network model developed in [4, 11] where the network is seen as a set of resources or links which are shared by a set of sources or users. The goal is to split bandwidth among sources in such a way that a social optimum is attained for all users sharing the network.

The problem of bandwidth allocation among flows reduces to finding the solution to the following concave optimization problem [4]:

$$\max_{x>0} \sum_{s\in S} U_s(x_s) \qquad , x = (x_1, ..., x_n),\ S = \{s_1, ..., s_n\} \quad (2)$$

$$\text{subject to: } \sum_{s\in S(l)} x_s \le c_l \qquad \forall l \in L$$

In this model the network is abstracted as a set of links $l \in L$ and each link l has the capacity c_l. The network is shared by sources $s \in S$ and each source s transmits data at rate x_s. When the source s sends data at rate x_s, it gets a utility $U_s(x_s)$ which is assumed to be a non-decreasing concave function twice differentiable. Also, let $S(l)$ denote the set of sources which use link $l \in L$ and $L(s)$ the set of links that source s uses.

Problem (2) is solved in a decentralized way using a dual approach. Let the Lagrangian for problem (2) be [11]

$$L(x,p) = \sum_{s \in S} U_s(x_s) - \sum_{l \in L} p_l \left(\sum_{s \in S(l)} x_s - c_l \right)$$

where p is the Lagrange multiplier associated with the inequality constraints of problem (2). p is a vector of prices p_l, one for each link l, where p_l is interpreted as the price per unit bandwidth at link $l \in L$. Then the objective function for the dual problem is (see [11] for a detailed derivation)

$$D(p) = \max_{x_s > 0} L(x,p) =$$
$$\sum_{s \in S} \max_{x_s > 0} (U_s(x_s) - x_s p^s) + \sum_{l \in L} p_l c_l$$
where $p^s = \sum_{l \in L(s)} p_l$

The dual problem is solved using a gradient-descend algorithm and after the dual optimal p is obtained, the bandwidth x_s that maximizes the Lagrangian is determined using the Karush-Kuhn-Tucker conditions [11]:

$$x_s(p^s) = U_s^{'-1}(p^s) \tag{3}$$

where $U^{'-1}$ is the inverse of $U_s^{'}$. In practice, we use as p^s the loss event rate of TFRC which is approximately dual optimal when Active Queue Management policies that stabilizes the queue length are deployed at the links.

3.2 The media-friendly control for video streaming

In order to make our control media-friendly we use in equation (3) an utility function which is suitable for layered scalable video streams and intuitively obeys the following rules:

- if the client prefetch buffer is small (e.g. smaller than a threshold), the utility should be high, because if we don't have a large enough throughput we might get an empty buffer at the client and stream playing can freeze;
- if the buffer is large then the utility should be small, but still it should follow the slope of the bitrate of the stream;

We have chosen the following utility function for video streaming applications:

$$U(x) = \frac{b}{b_{avg}} \left(1 + \frac{1}{\Delta} \right) log(x) \tag{4}$$

where x is the throughput of source, b is the bitrate measured in bytes for the current second of stream, b_{avg} is the average bitrate over the whole stream and Δ is the current value of the client prefetch buffer expressed in seconds. It is easy to see that $U(x)$ is concave with respect to x. In the expression of the utility function we considered only bitrate and client prefetch buffer values, but other media characteristics can be used too: bitrate averaged over a scene, quality measures, PSNR values etc. Using this utility function in equation (3) yields the following optimal bandwidth allocation:

$$x_u(p) = \frac{b}{b_{avg}} \left(1 + \frac{1}{\Delta} \right) \frac{1}{p} \tag{5}$$

where we use the notation x_u for the optimal bandwidth allocation to distinguish this later from the bandwidth allocated to the source by TFRC, x_{tfrc}, which is computed according to equation (1). This way, the control obtained in (5) is a media-friendly optimal congestion control, but it is not TCP-friendly.

3.3 Adding TCP-friendliness to the media-friendly control

We now have two congestion control algorithms: TFRC given by equation (1) which is TCP-friendly, but not media-friendly and the control given in (5) which is media-friendly and is the optimal solution of problem (2), but it is not TCP-friendly. We are looking for a control which is TCP-friendly and also media-friendly. In other words, we want a congestion control which gives a throughput evolution with a slope similar to the slope of TFRC's throughput and also similar to the slope of the throughput evolution given by control (5). Putting this into a mathematical form, yields:

$$X^{'}(t) = (1-\alpha)x^{'}_{tfrc}(t) + \alpha x^{'}_u(t)$$

where we considered the throughput evolution in time (i.e. all derivatives are taken with respect to t). $\alpha \in [0,1]$ is a normalized factor for controlling the trade-off between the TCP-friendly shape and media-friendly shape of the throughput. We suggest a value of 20 % for α to favour TCP-friendliness over media-friendliness of the congestion control.

Integrating the above relation with respect to the time, t, gives:

$$X(t) = (1-\alpha)x_{tfrc}(t) + \alpha x_u(t) + k$$

where k is a scaling factor. So, the final form of our TCP-friendly and media-friendly congestion control is:

$$X(p) = (1-\alpha)x_{tfrc}(p) + \alpha x_u(p) + k \tag{6}$$

where p is the congestion price from the dual problem.

4. FUTURE WORK

This paper describes work in progress that is currently in the implementation and testing phase. We developed a congestion control algorithm that is TCP-friendly and media-friendly. Such a congestion control is more suitable for video streaming applications than the classical TCP congestion control and smooth TCP-friendly rate controls because by taking into account media characteristics it should maximize the perceived quality of the video at client side in the given network conditions. It remains for us to test the congestion control given by equation (6) in real streaming scenarios and see that indeed it optimizes the delivery of video streams in best-effort networks.

Also, it is important to devise a dynamical rule for computing the scaling factor k so that our congestion control does not became overly media-friendly at some point and affect its TCP-friendliness or the opposite and this dynamical rule should fit a large distribution of TFRC and media-friendly throughputs.

Note that because we included TCP-friendliness into control (6), this congestion control approximates the optimal solution of problem (2), but is not the optimal solution. We intend to develop an equation for choosing the values of α so that the control approaches better the optimal solution of problem (2) while it remains TCP-friendly.

5. REFERENCES

[1] S. Floyd, M. Handley, J. Padhye, J. Widmer, *TCP Friendly Rate Control*, RFC 3448, January 2003.

[2] S. Floyd, M. Handley, J. Padhye, J. Widmer, *Equation-Based Congestion Control for Unicast Applications*, ACM SIGCOMM 2000.

[3] J. Padhye, V. Firoiu, D. Towsley, J. Kurose, *Modeling TCP Throughput: A Simple Model and its Empirical Validation*, ACM SIGCOMM 1998, Vancouver.

[4] F. P. Kelly, A. K. Maulloo, and D. K. H. Tan, *Rate control for communication networks: Shadow prices, proportional fairness and stability*, J. Oper. Res. Soc., vol. 49, no. 3, pp. 237-252, Mar. 1998.

[5] K. Kar, S. Sarkar, and L. Tassiulas, *A simple rate control algorithm for maximizing total user utility*, in Proc. IEEE INFOCOM, Apr. 2001, pp. 133-141.

[6] F. Paganini, Z. Wang, J. C. Doyle and S. H. Low, *Congestion control for high performance, stability and fairness in general networks*, IEEE/ACM Transactions on Networking, 13(1), pp.43-56, Feb. 2005.

[7] S. Kunniyur and R. Srikant, *End-to-End congestion control schemes: Utility functions, random losses and ECN marks*, IEEE/ACM Transactions on Networking, vol. 11, no. 5, pp. 689-702, Oct. 2003.

[8] F. Pereira, T. Ebrahimi, *The MPEG-4 Book*, Prentice Hall PTR, ISBN 0130616214, 9780130616210, 2002.

[9] G. Vinnicombe, *On the stability of networks operating TCP-like protocols* , in Proc. IFAC, Aug. 2002.

[10] Y. Zhang, S. R. Kang, and D. Loguinov, *Delay-Independent Stability and Performance of Distributed Congestion Control*, IEEE/ACM Transactions on Networking, vol. 15, no. 5, pp. 838-851, Oct. 2007.

[11] S. H. Low and D. E. Lapsley, *Optimization flow control I: Basic algorithm and convergence*, IEEE/ACM Transactions on Networking, vol. 7, pp. 861-874, Dec. 1999.

[12] W-H. Wang, M. Palaniswami and S. H. Low, *Application-Oriented Flow Control: Fundamentals, Algorithms and Fairness*, IEEE/ACM Transactions on Networking, 14(6), pp. 1282-1291, Dec. 2006.

[13] J. Yan, K. Katrinis, M. May, B. Plattner, *Media- and TCP-Friendly Congestion Control for Scalable Video Streams*, IEEE Transactions on Multimedia, Vol. 8, No. 2, April, 2006.

[14] R. Srikant, *The Mathematics of Internet Congestion Control*, Cambridge, MA: Birkhauser, 2004.

[15] S. Floyd, K. Fall, *Promoting the Use of End-to-End Congestion Control in the Internet*, IEEE/ACM Transactions on Networking, 7(4), pp. 458-472, Aug. 1999.