STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **LII**, Number 2, 2007

TOWARDS AN UTILITY-BASED TCP-FRIENDLY RATE CONTROL

STERCA ADRIAN

Abstract. We present a method for constructing TCP-friendly rate controls that are at the same time media-friendly. These types of rate controls are more suitable for multimedia streaming application than the classical TCP rate control. The method is developed by combining the notion of TCP-friendliness with a general optimization framework for bandwidth sharing in computer networks.

1. TCP-Friendliness

During the years 1980 when the Internet (Arpanet) grew from tens of computers to thousand of computers, Internet researchers noticed that Internet's core transport protocol, namely TCP, can not handle in an efficient manner the growing number of connections. Because the process of sharing bandwidth amoung users was not strongly regularized at the TCP level, the phenomenon of congestion collapse occured, reducing drastically the utility of the network. To avoid the occurence of congestion collapse, TCP incorporated Jacobson's AIMD (*Additive Increase Multiplicative Decrease*) congestion control algorithm [9] which increases the send rate by one packet per RTT in the absence of congestion indications and decreases the send rate by half when congestion does occur. Chiu and Jain proved in [1] that a simple AIMD congestion control algorithm like the one employed by TCP converges to a fair and efficient equilibrium state when the congestion feedback is received at the same time by all flows sharing the network and all flows react to it together synchronously. The fairness criterion towards which an AIMD algorithm converges, in the aforementioned conditions, is *max-min fairness* [4]. However, in real world conditions, different flows don't react to congestion synchronously and don't receive network feedback in the same time. Consequently, TCP

Received by the editors: October 20, 2007.

2000 *Mathematics Subject Classification.* 90B18, 68M20.

1998 *CR Categories and Descriptors.* 90B18 [**Operations research, mathematical programming**]: Operations research and management science – *Communic ation networks*; 68M20 [**Computer science**]: Computer system organization – *Performance evaluation; queueing; scheduling*;

doesn't reach maximum efficiency in practice and is approximately fair only across flows having the same RTT and the same congestion measure which doesn't happen in the real world.

Another significant characteristic of TCP is that it treats flows having the same RTT 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 in the sending bitrate, a variation that is not well coped with by current codecs. Steep degradations in the sending bitrate of a multimedia stream has very bad consequences on the quality perceived by the final receiver.

In an effort to steer the development of a congestion control mechanism for multimedia streaming, the scientific community has advertised the notion of TCP-friendly flow [6] as a flow which receives, on average, approximately the same bandwidth as a TCP flow under the same network traffic conditions. When the packet loss rate, p, is smaller than 0.3, the transmission rate of such a TCP-friendly flow should approximately be [6]

$$X = \frac{1.5 * \sqrt{\frac{2}{3}}}{RTT * \sqrt{p}} \text{ packets/second} \tag{1}$$

where RTT is the round-trip time and p is the packet loss rate this flow sees. [7] presents an equation which characterizes more accurately the throughput of a TCP flow, because it takes into account retransmission timeouts and doesn't restrict p to values smaller than 0.3. However we do not use this equation in our study because it is difficult to invert it.

2. Rate pricing

A different approach in sharing bandwidth among competing applications is taken in [2,3,4] where each application has a bandwidth utility function and bandwidth sharing is done in such a way that it maximizes the sum of all users' utility functions. The problem of bandwidth allocation among flows reduces to finding the solution to the following concave optimization problem:

$$(2) \qquad \begin{cases} \max_{x>0} \sum_{s \in S} U_s(x_s) & x = (x_1, ..., x_n), S = \{s_1, ..., s_n\} \\ \text{subject to: } \sum_{s \in s(l)} x_s \leq c_l & \forall l \in L \end{cases}$$

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 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 hard to solve in a decentralized way because of the coupling of transmit rates of sources at links in the inequality constraints of the problem. Instead of looking at this problem, the dual problem is considerred. Let the Lagrangian for problem (2) be [3]

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

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 $l \in L$. Because the first term in the Lagrangian is separable in x_s, so we have

$$\max_{x_s>0} \sum_{s \in S} \left(U_s(x_s) - x_s \sum_{l \in L(s)} p_l \right) = \sum_{s \in S} \max_{x_s>0} \left(U_s(x_s) - x_s \sum_{l \in L(s)} p_l \right)$$

the objective function of the dual problem is [3]:

$$(3) \qquad D(p) = \max_{x_s>0} L(x,p) = \sum_{s \in S} B_s(p^s) + \sum_{l \in L} p_l c_l$$

$$\text{where} \quad \begin{aligned} &B_s(p^s) = \max_{x_s>0}(U_s(x_s) - x_s p^s) \\ &p^s = \textstyle\sum_{l \in L(s)} p_l \end{aligned}$$

Applying the Karush-Kuhn-Tucker theorem to find x_s which maximizes the Lagrangian $L(x,p)$, the solution is [3]:

$$(4) \qquad x_s(p^s) = U_s^{'-1}(p^s)$$

where $U^{'-1}$ is the inverse of $U_s^{'}$. For x_s from (4) to be the unique maximizer that solves problem (2), p must be a Lagrange multiplier that satisfies the complementary slackness condition [5, prop. 3.3.4]. In practice, we use as p^s the loss event rate of TCP which satisfies with approximation the complementary slackness condition.

3. Mixing TCP-Friendliness with Rate pricing

We present in this section a method for finding a congestion control algorithm suitable for multimedia streaming applications by combining the TCP-friendly model with the Rate pricing model. More specifically, we first **a) derive the utility function of the system which is maximized by the solution of the TCP-friendly equation (1)**, then we **b) modify this utility function we have obtained to be more media specific (or media-friendly)** and then we **c) compute backwards using relation (4) the solution to the new optimization system**. In the rest of our calculus we will use equation (1) for characterization of TCP-friendliness and not the equation proposed by [7] because it is very difficult to invert the latter. To get an ideea of the dificulties involved in inverting TCP's equation from [7] please see [8].

In order to derive the utility function of the optimization system for which the TCP-friendly equation is a solution, we equalize the TCP-friendly equation (1) with the equation of the optimization system's solution, i.e. equation (4),

$$x_s(p) = \frac{1.5 * \sqrt{\frac{2}{3}}}{RTT * \sqrt{p}} = U_s^{'-1}(p^s)$$

By inverting the function from the right-hand side of the equation, we get

$$U_s^{'}(x_s) = \frac{1.5^2 * \frac{2}{3}}{RTT^2 * x_s^2}$$

and then

$$(5) \quad U_s(x_s) = \int \frac{1.5^2 * \frac{2}{3}}{RTT^2 * x_s^2} dx = -\left(\frac{1.5 * \sqrt{\frac{2}{3}}}{RTT} \right)^2 * \frac{1}{x_s} + k \quad \text{,k is a constant}$$

is TCP's utility function. By maximizing the utility function presented above we obtain *weighted minimum potential delay fairness* [4].

In the second step of our method, we modify TCP's original utility function obtained in (5) to be more media specific. We consider two versions of media specific utility function based on (5):

$$U_s(x_s) = -\frac{b}{b_{avg}} * \left(\frac{1.5 * \sqrt{\frac{2}{3}}}{RTT}\right)^2 * \frac{1}{x_s} \tag{6}$$

$$U_s(x_s) = -\left(\frac{1.5 * \sqrt{\frac{2}{3}}}{RTT}\right)^2 * \frac{\sqrt{x}}{x_s} \tag{7}$$

where b is the multimedia stream's bitrate in the last second and b_{avg} is the multimedia stream's average bitrate.

If we solve using equation (4) the new optimization systems corresponding to the two utility functions depicted above, we get the following solutions:

$$x_s(p) = \frac{b}{b_{avg}} * \frac{1.5 * \sqrt{\frac{2}{3}}}{RTT * \sqrt{p}} \tag{8}$$

and

$$x_s(p) = \sqrt[3]{\frac{1.5^2 * \frac{2}{3}}{4 * RTT^2 * p^2}} \tag{9}$$

The utility function from (6) is media-friendly because it takes into account the bitrate demands of the stream (i.e. when the instant bitrate is above the average bitrate the application has a higher utility of bandwidth x_s) and the utility function from (7) is also useful for multimedia streaming applications because it tries to reduce fluctuations on the transmision rate x_s.

4. Conclusions and Future work

In this paper we have developed a method for obtaining rate controls that are TCP-friendly and in the same time media-friendly. We exemplified this method by two such rate control algorithms (equations (8) and (9)) which maximize an application-specific utility function. As future work we intend to test the two rate controls we developed here in varying network environments and to find more appropriate utility-functions for multimedia specific applications and use them to develop improved TCP-friendly and media-friendly rate controls.

REFERENCES

- [1] Chiu, D. M., Jain, R., *Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks*, Computer Networks and ISDN Systems 17, North-Holland, 1989.
- [2] Kelly, F., Maulloo, A., Tan, D., *Rate control in communication networks: shadow prices, proportional fairness and stability*, Journal of the Operational Research Society 49 (1998), pp. 237-252.
- [3] Low, S., Lapsley, D. E., *Optimization Flow Control I: Basic Algorithm and Convergence*, IEEE/ACM Transactions on Networking, Vol. 7, No. 6, Dec. 1999.
- [4] Srikant, R., *The Mathematics of Internet Congestion Control*, Birkhauser, 2004.
- [5] Bertsekas, D., *Nonlinear Programming*, Athena Scientific, 1995.
- [6] Floyd, S., Fall, K., *Promoting the Use of End-to-End Congestion Control in the Internet*, IEEE/ACM Transactions on Networking, August 1999.
- [7] Padhye, J., Firoiu, V., Towsley, D., Kurose, J., *Modeling TCP Throughput: A Simple Model and its Empirical Validation*, ACM SIGCOMM 1998, Vancouver.
- [8] Sterca, A., *Finding TCP's utility function*, Technical Report, 2007 (available at http://cs.ubbcluj.ro/ forest/phd/Technical%20Reports).
- [9] Jacobson, V., *Congestion Avoidance and Control*, SIGCOMM Symposium on Communications Architectures and Protocols, pp. 314-329, 1988.

BABES-BOLYAI UNIVERSITY, FACULTY OF MATHEMATICS AND COMPUTER SCIENCE, DEPARTMENT OF COMPUTER SCIENCE, CLUJ-NAPOCA

E-mail address: `forest@cs.ubbcluj.ro`