PERIODICA POLYTECHNICA SER. EL. ENG. VOL. 45, NO. 2, PP. 79–90 (2001)

BANDWIDTH SHARING SCHEME OF END-TO-END CONGESTION CONTROL PROTOCOLS

Dung Dinh LUONG and József BÍRÓ

High Speed Networks Laboratory
Department of Telecommunications and Telematics
Budapest University of Technology and Economics
H–1521 Budapest, Hungary
{luong,biro}@ttt-atm.ttt.bme.hu

Received: May 22, 2002

Abstract

In a general network, it is not easy to find according to which criterion the available bandwidth would be shared between competing flows. In this paper, we propose a technique to find the bandwidth sharing scheme of end-to-end congestion control protocols. This technique divides the analysis work into two separate steps. In the first step, one should find the rate function, which expresses the relation between the throughput and the congestion measure. In the second step, the utility function can be obtained from the rate function.

Keywords: bandwidth sharing scheme, global optimization, TCP Reno, TCP Vegas.

1. Introduction

Up to the present, there exist no general methods to figure out in which way the end-to-end congestion control protocols such as TCP share the network resources between end-users.

F. KELLY [8] has recently introduced the *utility-optimization-based fairness*, which comes from a newly found economic theory. Rate distribution according to utility-optimization-based fairness must maximize the *objective function* representing the overall utility of the flows in the network.

Using this approach, researchers have been trying to characterize the bandwidth sharing scheme of existing congestion control protocols. M. VOJNOVIC et al. have found out that in the networks employing additive increase/multiplicative decrease control mechanism, the bandwidth is distributed in order to maximize a utility function called F_A^h [21]. They called this criterion the F_A^h *fairness*. On the other hand, S. H. LOW et al. have proved that TCP Vegas conforms to the weighted proportional fairness, which has the logarithm utility function [13]. By another approach, D. LUONG et al. have received the same result about the proportional fairness of TCP Vegas [14]. Despite the findings above, no systematic general method has been found to make the analysis work of bandwidth sharing easier.

In this paper, we propose a technique to find the bandwidth sharing scheme of end-to-end congestion control protocols. This technique divides the analysis work into two separate steps. In the first step, one should find the rate function, which expresses the relation between the throughput and the congestion measure. In the second step, the utility function can be obtained from the rate function.

The rest of this paper is organized as follows. Section 2 describes utility-optimization-based fairness. In Section 3 the congestion measures are generalized and the relation between congestion measure and global fairness optimization is presented. Section 4 describes the proposed technique and Section 5 demonstrates two application examples. The exactness of the bandwidth sharing scheme found in Section 5 is investigated in Section 6. We conclude this paper in Section 7.

2. Fairness Criteria in Terms of Global Optimization

2.1. The Network Model

The following network model has been used for introducing the utility-optimization-based fairness criteria [8, 7, 11, 12, 17, 9, 6]. Let's consider a network with a set of links L. Each link $l \in L$ has a finite capacity $\mu_l > 0$. Let F denote the set of flows which have data to transfer over the network, each flow $f \in F$ goes through a non-empty subset of L, called a route. We define an indicator function

$$I(f,l) = \begin{cases} 1, & l \in \text{the route of flow } f, \\ 0, & \text{otherwise.} \end{cases}$$

Let λ_f be the bandwidth allocation of the flow f. Any feasible bandwidth allocations must satisfy the *capacity constraint*

$$\sum_{f \in F} I(f,l)\lambda_f \leq \mu_l, \qquad l \in L. \tag{1}$$

The most common rate sharing scheme for competing flows is that the bandwidth should be shared as equally as possible and this results in the *max-min fairness*. In spite of the fact that the notion of fairness in resource sharing usually indicates the max-min fairness, there is no economic motivation for this criterion.

F. Kelly [8] argues that bandwidth should rather be shared in such a way to maximize an objective function representing the overall utility of the flows in the network. Each flow f with rate allocation λ_f has a utility $U_f(\lambda_f)$, which is an increasing, strictly concave and continuously differentiable function of λ_f [20]. The overall utility is assumed to be additive, meaning that it is $\sum_{f \in F} U_f(\lambda_f)$. The rate sharing scheme under this model is the solution of the following optimization problem

P:

$$\begin{aligned}&\text{maximize } \sum_{f\in F} U_f(\lambda_f)\\&\text{subject to } \sum_{f\in F} I(f,l)\lambda_f \le \mu_l \text{ for all } l \in L\\&\text{over } \lambda_f \ge 0 \text{ for all } f \in F\end{aligned}$$

It was proved in [7] that in the case of finite sets L and F, the solution vector of bandwidth shares exists and is unique.

2.2. Some Fairness Criteria

A bandwidth allocation $(\lambda_f,\ f \in F)$ is *proportionally fair* if it maximizes $\sum_{f\in F} \log \lambda_f$ while keeping the capacity constraints. In other words, *the proportional fairness* maximizes the overall utility of all flows, which utility function is logarithmic.

In general case, each flow f can be assigned with a weight ϕ_f, then a feasible rate vector $\underline{\lambda}$ is *weighted proportionally fair* if it maximizes $\sum_{f\in F} \phi_f \log \lambda_f$.

3. Congestion Measure

3.1. The Generalization of the Congestion Measure

In end-to-end congestion control protocols, sources continuously obtain feedback from the network, detect the level of congestion along their network path and adjust their sending rate accordingly. The level of congestion can be represented by the loss rate in TCP Tahoe/Reno or marking probability in ECN-capable protocols. We call them the *congestion measures*.

Link l is the *congested link* if the aggregate source rate reaches its capacity. That is, the inequality symbol is replaced by the equality symbol in (1). In a lossy network, we assume that the possible lost portion of the traffic is negligible. So the aggregate rate on a congested link is considered to be equal to the link's capacity even though losses could happen.

The congestion control mechanisms always try to use all of the available bandwidth. Therefore, any flow must have at least one congested link on its network path. Note that the congested link should not be confused with the *bottleneck* link of a network path, which has smallest capacity on that path.

Denote s_l be the level of congestion measure at link l. We assume that s_l is not negative. Furthermore, s_l is equal to 0 if l is not a congested link. These assumptions are consistent with all types of congestion measure mentioned above. Let s_f be the level of end-to-end congestion measure experienced by flow f. In the rest of this paper we call the level of congestion measure simply the *congestion level*.

We assume that the end-to-end congestion control protocol applied by the flows converges and stabilizes. In the stable state, each flow f has the rate of λ_f

$$\lambda_f = R_f(s_f), \tag{2}$$

where R_f is a strictly decreasing function and has positive values if s_f is positive.

Definition 1 A congestion measure is additive if the end-to-end congestion level experienced by a flow is equal to the sum of congestion levels at all links along the path of that flow.

$$s_f = \sum_{l \in L} I(f,l)s_l \quad \text{for all} \quad f \in F. \tag{3}$$

Obviously, the queuing delay, used in J. MO's protocol [17], is an additive congestion measure since end-to-end queuing delay is the sum of queuing times at relevant routers. Similarly, the link's price, used in S. H. LOW's protocol, is also additive and clearly proved in Page 2 of [2]. However, the loss rate, used in TCP Tahoe/Reno or in protocol proposed by S. KUNNIYUR [9], is definitely not additive. Nevertheless, in Section 5.2.1 we will replace the loss rate by another measure which is additive.

3.2. The Relation between Congestion Measure and Global Fairness Optimization

Proposition 1 *If a congestion control protocol stabilizes and the relevant congestion signal is additive, then the rate sharing of that protocol is the global solution of the optimization problem P if and only if the rate function is* $U_f^{'-1}$.

Proof In the stable state, if we have

$$\lambda_f = U_f^{'-1}(s_f), \tag{4}$$

then

$$U_f^{'}(\lambda_f) = s_f = \sum_{l \in L} I(f,l)s_l, \tag{5}$$

where $s_l \geq 0$ for all l. Furthermore, $s_l = 0$ if the aggregate source rate at link l is strictly less than its capacity $\sum_{f \in F} I(f,l)\lambda_f < \mu_l$.

The conditions described above are the Kuhn–Tucker conditions for constrained optimization problems. Because the utility functions are strictly concave and the constraint equations are linear, the Kuhn–Tucker conditions are necessary and sufficient for a global maximum [19]. □

4. The Description of the Technique

The technique under consideration divides the analysis of bandwidth sharing into two separate tasks:

1. **Finding rate function:**
   Let's consider the network as a black-box which continuously sends feedback indicating network congestion level to the sender. The operation of end-to-end congestion control protocols totally depends on that congestion measure. Hence, the throughput of a flow controlled by such a protocol is a function of congestion measure. The aim of this stage is to find the rate function. If the relevant congestion measure is not additive, try to find another measure which is additive.
2. **Finding utility function:**
   Let R_f be the rate function of flow f, found in the first task. Assume that $R_f(s)$ is a strictly decreasing function and has positive values if s is positive. According to Theorem 1, the utility function of flow f will be:

$$U_f(\lambda) = \int R_f^{-1}(\lambda) d\lambda. \tag{6}$$

Remark 1 *If $R_f^{-1}(\lambda)$ has a positive value when λ is positive, then the utility function $U_f(\lambda)$ from (6) is an increasing, strictly concave and continuously differentiable function of λ_f.*

5. Application Examples

5.1. Fairness of TCP Vegas

5.1.1. Rate Function of TCP Vegas

The congestion avoidance of TCP Vegas is clearly described in [3, 5, 10, 4]. Let *BaseRTT* be the minimum of all observed round-trip times, then *BaseRTT* is the estimate of the round-trip time without queuing delay. The expected and actual throughput is given by

$$Expected = \frac{W}{BaseRTT}; \quad Actual = \frac{W}{RTT}.$$

where W is the size of the current *cwnd*. Because *BaseRTT* is the minimum of *RTTs*, the actual throughput cannot be larger than the expected throughput. TCP Vegas uses the difference between them in congestion avoidance decision.

$$diff = Expected - Actual.$$

TCP Vegas makes decision according to the following

$$W_{next} = \begin{cases} W_{cur} + 1, & diff.BaseRTT < \alpha, \\ W_{cur} - 1, & diff.BaseRTT > \beta, \\ W_{cur}, & \alpha \le diff.BaseRTT \le \beta, \end{cases}$$

where W_{cur} is the current and W_{next} is the next size of *cwnd*. Let T_q denote the total queuing delay, which is equal to the difference between the actual *RTT* and *BaseRTT*. We rewrite the quantity *diff.BaseRTT* using T_q:

$$diff.BaseRTT = \frac{T_q}{RTT} W. \tag{7}$$

For the f flow in the stable state, we have from (7)

$$\alpha \le \frac{T_{qf}}{RTT_f} W_f \le \beta, \tag{8}$$

where T_{qf}, RTT_f, W_f are, respectively, the end-to-end queuing delay, round-trip time and congestion window size of f. Note that the actual rate λ_f is equal to the window size divided by the round-trip time. Then we have

$$\frac{\alpha}{T_{qf}} \le \lambda_f \le \frac{\beta}{T_{qf}}. \tag{9}$$

Eq. (9) infers that the rate share has a convergence range and can take any value from that range [14]. For simplicity, we assume that TCP Vegas always has the rate share which is the middle point of the convergence range. We can approach this assumption by set α and β close to each other. Then the rate function will be:

$$R_f(s_f) = \frac{\gamma}{T_{qf}}; \gamma = \frac{\alpha + \beta}{2}. \tag{10}$$

It is clear that the queuing time is an additive measure, since the end-to-end total queuing delay is equal to the sum of queuing time at all nodes lying on the network path.

5.1.2. *Utility Function of TCP Vegas*

From (10), we have

$$R_f^{-1}(\lambda) = \frac{\gamma}{\lambda}. \tag{11}$$

Then the utility function will be

$$U_f(\lambda) = \int R_f^{-1}(\lambda) d\lambda = \gamma \log(\lambda). \tag{12}$$

That means TCP Vegas conforms to the proportional fairness, defined in Section 2.2. This result is consistent with the contribution of [13] and [14]. We have to mention that in [13], TCP Vegas is proved to be weighted proportionally fair because each TCP flow uses different thresholds α and β while in [14] and in this paper, all flows have the same initial parameters.

5.2. Fairness of TCP Reno

5.2.1. Rate Function of TCP Reno

Assuming that losses are random, by different approaches [16, 18] researchers have obtained the throughput of TCP Reno as a function of loss rate. A complicated form of throughput formula was introduced by J. PADHYE et al. [18]. In this paper, we use the following simplest form [16] of throughput formula to infer the utility function

$$Throughput(f) = \frac{c_f MSS_f}{RTT_f \sqrt{p_f}}, \tag{13}$$

where MSS_f and RTT_f are the Maximum Segment Size and the Round Trip Time, p_f is the loss rate of flow f and c_f is a constant which depends on the current TCP implementation, the ACK strategy (delayed or not) [16]. As we have mentioned, the loss rate is not an additive congestion measure. Therefore, we will convert (13) into a function of additive congestion measure.

Denote ϕ_f be the probability that a packet of flow f is not lost. Similarly, let ϕ_l denote the probability that a packet is not lost on the link l. We have $\phi_f = 1 - p_f$ and $\phi_l = 1 - p_l$. Assuming that the loss process on a link is independent of that one on another link, we can write

$$\phi_f = \prod_{l \in L} \phi_l^{I(f,l)} \longrightarrow \log \phi_f = \sum_{l \in L} I(f,l) \log \phi_l. \tag{14}$$

Denote $s_f = -\log \phi_f$ and $s_l = -\log \phi_l$. Then s is an additive congestion measure and from (13) the rate function of TCP Reno will be

$$R_f(s_f) = \frac{C_f}{\sqrt{1 - e^{-s_f}}}; C_f = \frac{c_f MSS_f}{RTT_f}. \tag{15}$$

5.2.2. Utility Function of TCP Reno

From (15), we have

$$R_f^{-1}(\lambda) = -\log\left(1 - \frac{C_f^2}{\lambda^2}\right). \tag{16}$$

Then the utility function will be

$$U_f(\lambda) = \int R_f^{-1}(\lambda)d\lambda$$
$$= 2\lambda \log \lambda - (\lambda - C_f) \log(\lambda - C_f) - (\lambda + C_f) \log(\lambda + C_f). \quad (17)$$

We call the fairness with the utility function in (17) the F_b fairness, which differs from the F_a^h fairness found by M. VOJNOVIC et al. [21], with the utility function

$$U_f(\lambda) = \frac{1}{RTT_f} \log \left(\frac{\lambda}{\frac{1}{RTT_f} + 0.5\lambda} \right). \quad (18)$$

One should not be surprised at two different utility functions of (17) and (18). Since TCP Reno is a very complicated protocol, then it is hard to model it accurately. For instance, there are several rate functions founded for TCP Reno. The accuracy of a utility function depends on which rate function we have used.

6. Testing the Founded Fairness Criteria

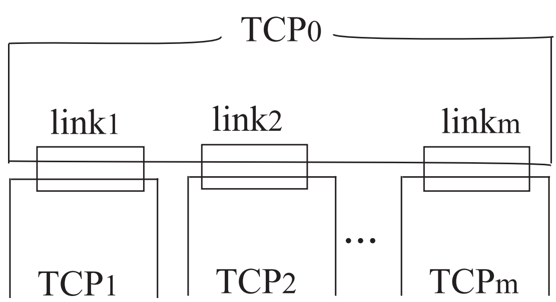

Fig. 1. Simulation network model for testing fairness

The widely used network model [15] is chosen for testing the fairness criteria of TCP Vegas and Reno (*Fig. 1*). The flow TCP_0 goes through m nodes, while $TCP_1, TCP_2, \ldots TCP_m$ pass only one link. All the links have the same bandwidth of 10 Mbps. The propagation delay and buffer size is set to 100 ms and 50 packets, respectively, for all links.

Let $\underline{\lambda}$ be the allocated rate vector according to the fairness criterion to be tested and $\underline{\lambda'}$ is the throughput vector, observed by simulator ns-2 [1]. $\underline{\lambda}$ has been calculated by solving the problem P using the Optimization Toolbox of Matlab. The round-trip time of each flow is needed for solving the problem P of F_a^h and F_b fairness criteria. They are observed via simulation.

The relative error based on the Euclidean norm is used to characterize the accuracy of the fairness criterion, or in other words, how close the fairness criterion to be tested is to the actual bandwidth sharing scheme in practice.

$$relative\ error = \frac{\|\underline{\lambda} - \underline{\lambda}'\|_2}{\|\underline{\lambda}'\|_2}, \tag{19}$$

where $\|\underline{x}\|_2$ is the the Euclidean norm of vector $\underline{x}$

$$\|\underline{x}\|_2 = \sqrt{\sum_i x_i^2}. \tag{20}$$

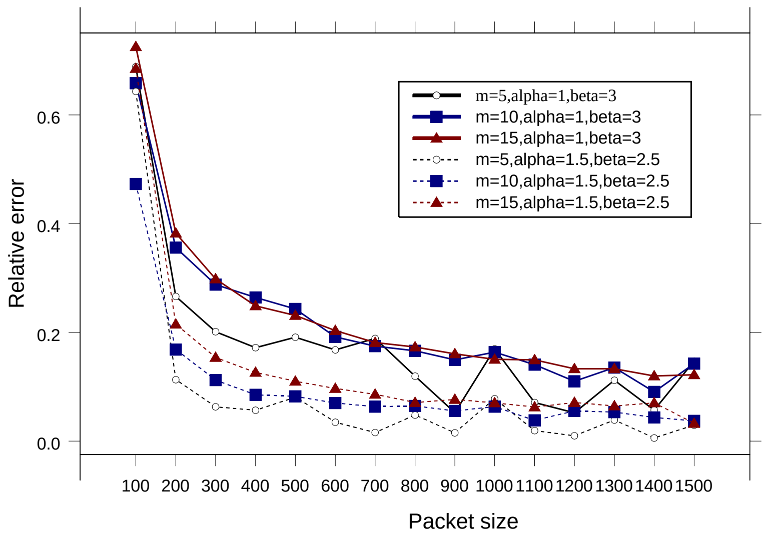

Fig. 2. The Euclidean error of the proportional fairness compared to TCP Vegas's rate sharing

Fig. 2 shows the relative errors of the proportional fairness compared to TCP Vegas's rate sharing versus the packet size. Three cases have been considered according to $m = 5, 10, 15$. We have modified the ns-2's implementation of TCP Vegas in such a way that α and β can have real positive values. The bold curves represent the errors when $\alpha = 1; \beta = 3$, while the thin curves are for $\alpha = 1.5; \beta = 2.5$. The smaller the network (smaller m) or the larger the packet size, the smaller the relative error is. Beside that, the relative error is smaller when α and β are set closer to each other ($\alpha = 1.5; \beta = 2.5$), compared to the case of $\alpha = 1; \beta = 3$. The reason is that the smaller difference between α and β makes the rate function of (10) more accurate.

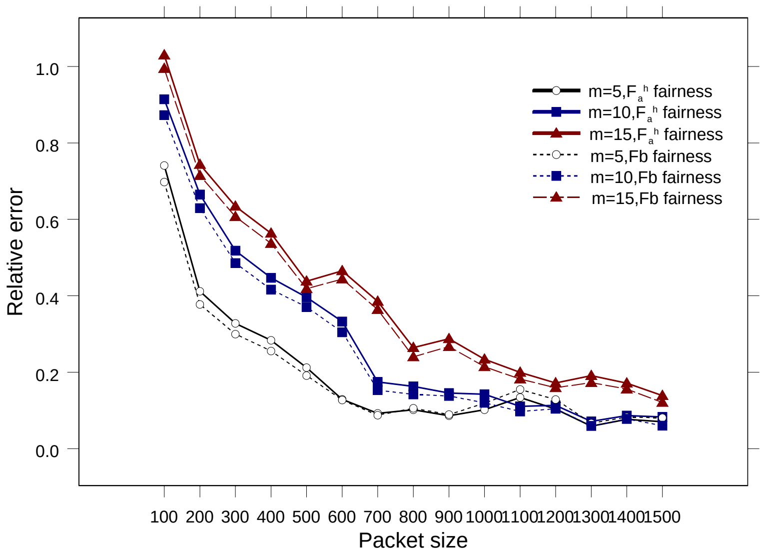

Fig. 3. The Euclidean error of the F_a^h and F_b fairness criteria compared to TCP Reno's rate sharing

Similarly, *Fig. 3* shows the relative errors of the F_a^h and F_b fairness criteria compared to TCP Reno's rate sharing versus the packet size. The F_b scheme noticeably has smaller relative error than the F_a^h fairness. This observation is an evidence to guess that F_b is closer to TCP Reno's real bandwidth sharing than F_a^h. In this case, the relative errors tend to increase when the packet size decreases or the network becomes larger (m increases).

After comparing *Fig. 2* with *Fig. 3*, one could conclude that the relative error in the case of TCP Vegas is smaller than in the case of TCP Reno. It may be inferred that the proportional fairness is closer to TCP Vegas's rate sharing than F_a^h or F_b fairness to TCP Reno. The reason must be found in the rate functions: it is not easy to find an accurate rate function for TCP Reno.

The decrease of the relative error caused by the increase of packet size is an interesting observation which we have not expected. The explanation for that phenomenon would require further research works.

7. Conclusion and Future Work

This paper gives a systematic general technique to gain the bandwidth sharing scheme for end-to-end congestion control protocols. The proposed technique divides the analysis of bandwidth sharing, which used to be a very complicated process, into two separate tasks. In the first task, one should find the rate of a flow as a function of congestion measure. In the second task, the utility function can be inferred from the rate function. With this technique, bandwidth sharing scheme is much easier to be obtained.

To demonstrate the effectiveness of the fairness finding technique, we show how to find the fairness of TCP Vegas and Reno. By simulation we show which factors the accuracy of the observed schemes depends on.

In a real network having complicated mechanisms, we should consider many further constraints besides the capacity constraint. These restrictions may have significant role in optimization. The investigation of the impact of other possible constraints on the rate sharing would be the future work.

References

[1] ns2: Network simulator. http://www-mash.cs.berkeley.edu/ns/ns.html.

[2] ATHURALIYA, S. – LOW, S. H., *Optimization Flow Control (ii): Implementation*. http://netlab.caltech.edu/pub/rem.htm.

[3] BRAKMO, L. S. – O'MALLEY, S. W. – PETERSON, L. L., TCP Vegas: New Techniques for Congestion Detection and Avoidance, In *Proceedings of ACM SIGCOMM'94*, London, UK, August 1994, pp. 24–35.

[4] HASEGAWA, G. – MURATA, M. – MIYAHARA, H., Fairness and Stability of Congestion Control Mechanisms of tcp, In *Proceedings of IEEE INFOCOM'99*, March 1999, pp. 1329–1336.

[5] HENGARTNER, U. – BOLLIGER, J. – GROSS, T., TCP Vegas Revisited. In *Proceedings of IEEE INFOCOM'2000*, Tel Aviv, Israel, March 2000.

[6] KAR, K. – SARKAR, S. – TASSIULAS, L., A Simple Rate Control Algorithm for Maximizing Total User Utility, In *Proceedings of IEEE INFOCOM'2001*, Los Alamitos, CA, April 2001, pp. 133–141.

[7] KELLY, F. P., Charging and Rate Control for Elastic Traffic, *European Transactions on Telecommunications*, **8** (1997), pp. 33–37.

[8] KELLY, F. P. – MAULLOO, A. K. – TAN, D. K. H., Rate Control in Communication Networks: Shadow Prices, Proportional Fairness and Stability, *Journal of the Operational Research Society*, **49** (1998), pp. 237–252.

[9] KUNNIYUR, S. – SRIKANT, R., End-to-end Congestion Control Schemes: Utility Functions, Random Losses and ECN Marks, In *Proceedings of the IEEE INFOCOM'2000*, Los Alamitos, March 2000, pp. 1323–1332.

[10] KURATA, K. – HASEGAWA, G. – MURATA, M., Fairness Comparisons between tcp Reno and tcp Vegas for Future Deployment of tcp Vegas, In *Proceedings of INET'2000*, January 2000.

[11] LA, R. – ANANTHARAM, V., Charge-Sensitive TCP and Rate Control in the Internet, In *Proceedings of IEEE INFOCOM'2000*, Los Alamitos, March 2000, pp. 1166–1175.

[12] LOW, S. H. – LAPSLEY, D. E., Optimization Flow Control I: Basic Algorithm and Convergence, *IEEE/ACM Transactions on Networking*, **7** No. 6, December (1999), pp. 861–874. http://www.acm.org/pubs/citations/journals/ton/1999-7-6/p861-low/.

[13] LOW, S. H. – PETERSON, L. L. – WANG, L., Understanding tcp Vegas: a Duality Model, *Journal of the ACM (JACM)*, **49** No. 2 (2002), pp. 207–235.

[14] LUONG, D. D. – BÍRÓ, J., On the Proportional Fairness of tcp Vegas, In *Proceedings of IEEE GlobeCom*, San Antonio, USA, November 2001.

[15] MASSOULIE, L. – ROBERTS, J., Bandwidth Sharing: Objectives and Algorithms, In *Proceedings of IEEE INFOCOM'99*, New York, NY, March 1999.

[16] MATHIS, A. M. – SEMKE, J. – MAHDAVI, J. – OTT, T., The Macroscopic Behavior of the tcp Congestion Avoidance Algorithm, *Communication Review*, **27**, July 1997.

[17] MO, J. – WALRAND, J., Fair End-to-End Window-Based Congestion Control, *IEEE/ACM Transactions on Networking*, **8** No. 5, October (2000), pp. 556–567.

[18] PADHYE, J. – FIROIU, V. – TOWSLEY, D. – KRUSOE, J., Modeling TCP Throughput: A Simple Model and its Empirical Validation, *Proceedings of the ACM SIGCOMM'98* August 1998, pp. 303–314.

[19] PIKE, R. W., *Optimization for Engineering Systems*, Van Nostrand Reinhold Company Inc., New York, 1986.

[20] SHENKER, S., Fundamental Design Issues for the Future Internet, *IEEE J. Selected Areas in Communications*, September 1995.

[21] VOJNOVIC, M. – LE BOUDEC, J.-Y. – BOUTREMANS, C., Global Fairness of Additive-Increase and Multiplicative-Decrease with Heterogeneous Round-Trip Times, In *Proceedings of IEEE INFOCOM'2000*, Los Alamitos, March 2000, pp. 1303–1312.