

How expensive is link utilization?

Rade Stanojević and Robert Shorten

Hamilton Institute, NUIM, Ireland

Abstract. Understanding the relationship between queuing delays and link utilization for general traffic conditions is an important open problem in networking research. Difficulties in understanding this relationship stem from the fact that it depends on the complex nature of arriving traffic and the problems associated with modelling such traffic. Existing AQM schemes achieve a “low delay” and “high utilization” by responding early to congestion without considering the exact relationship between delay and utilization. However, in the context of exploiting the delay/utilization tradeoff, the optimal choice of a queueing scheme’s control parameter depends on the cost associated with the relative importance of queuing delay and utilization. The optimal choice of control parameter is the one that maximizes a benefit that can be defined as the difference between utilization and cost associated with queuing delay. We present a generic algorithm Optimal Delay-Utilization control of t (ODU- t) that is designed with a performance goal of maximizing this benefit. Its novelty lies in fact that it maximizes the benefit in an online manner, without requiring knowledge of the traffic conditions, specific delay-utilization models, nor does it require complex parameter estimation. Moreover, other performance metrics like loss rate or jitter can be directly incorporated into the optimization framework as well. Packet level ns2 simulations are given to demonstrate the behavior of the proposed algorithm.

1 Introduction

Current router buffers are generally sized by the rule-of-thumb given in [25]: router buffers require approximately space for $B = \overline{RTT} \times C$ packets, where \overline{RTT} is the “average” round trip time for connections that use the link and C is capacity of the link. Following this rule, most router buffers are designed in such a fashion that they result in up to $100ms$ to $250ms$ of queueing [1, 4]. This, together with TCP’s mechanism of congestion avoidance, serves to ensure a high link utilization.

In the last few years a number of results related to buffer sizing for congested links have appeared [1, 2, 4, 5] that suggest significantly smaller buffers. Although the bounds from these papers yield important theoretical insights into the relation between link utilization and the required buffering they are not immediately applicable to buffers in the real Internet routers for a number of reasons. Firstly, these bounds are functions of various parameters such as the number of active users that are bottlenecked at the link; RTT distribution of TCP users, TCP

parameters ($maxcwnd_$), etc. These quantities vary, and are also usually very hard to estimate [6, 14, 13, 24, 3]. Secondly, the mathematical assumptions used in deriving of these bounds are quite restrictive and do not take into account the various and variable traffic mixes possible, the level of loss synchronization, the existence of non-TCP traffic, etc. Most importantly, while it is useful to know that delay and utilization are related in some manner, it is not immediately clear how to utilize this relationship in a meaningful manner.

In this paper we build an optimization framework for the design of queue management schemes in which (low) queueing delays are considered as a scarce resource together with link utilization. The relative importance between queueing delays and utilization is a user¹ specified parameter. Therefore the optimal choice of queueing scheme parameter t is one that maximizes overall benefit $B(t)$ that takes into account the relative importance of queueing delays and utilization. Queueing scheme parameter t can be: (1) available DropTail queue space, or (2) per packet drop probability, or (3) virtual queue service rate or any other parameter that can control utilization and queueing delays.

In Section 3 we propose an online algorithm for control of generic parameter t : Optimal Delay-Utilization control of t (ODU- t). It does not require intricate measurement techniques neither specific assumptions related to the nature of traffic mix. In Section 4 we present a brief simulation study of ODU- t for t denoting available DropTail space.

2 Optimization framework

Let us consider a synthetic example in which average queueing delay ($aQD(t)$) and utilization $u(t)$ depend on choice of queueing parameter t given in Table 1. For simplicity, assume for the moment that the parameter t is the available buffer space on the congested FIFO Drop-Tail queue; for buffer size equal to t_1 the average queue delay is $100ms$ and the utilization is 100%, for buffer size equal to t_2 the average queue delay is $20ms$ and the utilization is 98%, and so on. Which choice of t is optimal (among 4 possible in this example), depends on the “importance” of low queueing delays. To formalize this, one can identify the “importance” by the relative price between utilization and queueing delays. Let $P : [0, \infty) \mapsto [0, \infty)$ be a function that specifies relative price between utilization and delays. In other words, a queueing delay of d seconds has same value as utilization of $P(d)$. Formally, a price function is any function that satisfies the following definition.

Definition 1. *The function $P : [0, \infty) \mapsto [0, \infty)$ is a price function if it is twice differentiable, increasing and convex. In other words if:*

- (a) $\forall d \in [0, \infty) \exists P''(d)$
- (b) $\forall d \in [0, \infty) P'(d) \geq 0$
- (c) $\forall d \in [0, \infty) P''(d) \geq 0$

¹ In this context user is an ISP or link owner.

t	t_1	t_2	t_3	t_4
$aQd(t)(sec)$	0.1	0.02	0.005	0.001
$u(t)$	1.00	0.98	0.90	0.60

Table 1. Synthetic example of $aQd(t)$ and $u(t)$ for 4 different possible choices of parameter t .

Having defined a price function, the overall benefit (in the case given by the parameter t) can be written in the form:

$$B(t) = u(t) - P(aQd(t)). \quad (1)$$

Comment. Notion, similar to the benefit $B(t)$ is introduced in [2] for t representing the available buffer size.

The definition of benefit allows us to define a notion of optimal choice, as the value of t that maximizes the benefit. Formally:

Definition 2. For a given price function P and set \mathcal{T} of possible choices of t , an optimal Delay-Utilization(D-U) choice is any t_0 such that

$$B(t_0) = \max\{B(t) \mid t \in \mathcal{T}\}, \quad (2)$$

if the maximum on the right hand side exists.

In the example given in Table 1, if we completely ignore the importance of low queueing delays, by setting $P(d) \equiv 0$ for all d , then the optimal D-U choice is given by t_1 , as this maximizes the benefit $B(t) = u(t) - P(aQd(t)) = u(t)$ on the set $\mathcal{T} = \{t_1, t_2, t_3, t_4\}$. For the price function $P(d) = 5 \cdot d$, the optimal D-U choice is t_2 , and for the price function $P(d) = 100 \cdot d$, the optimal D-U choice is t_4 .

Throughout this paper we assume:

Assumption 1 Under static traffic conditions the overall benefit given by (1) is a concave function of t .

Assumption 1 is very hard to analytically check. In a theoretical framework, this would require accurate models of various traffic mixes, and as we already noted, modelling such complex environments is highly nontrivial. Some results related to the convex relationship between utilization and buffer size in non-elastic traffic environments are developed in [17, 18]. However, our empirical observations suggest that for the traffic mix that is consisted from the static number of TCP and UDP flows, Assumption 1 holds when t is (1) the available DropTail space or (2) per-packet drop probability or (3) Virtual queue service rate. Technical report [22] discuss this in more detail, and contains the packet level ns2 simulations results that validate Assumption 1 under mentioned circumstances.

3 Optimal Delay-Utilization control of t

Convex optimization has been widely employed in the networking community see [15, 23]. In our case, we need an efficient algorithm for solving (2). A standard control strategy for solving (2) is given by

$$t(k+1) = t(k) \left(1 + g(k) \frac{B(t(k)) - B(t(k-1))}{t(k) - t(k-1)} \right), \quad g(k) \geq \epsilon > 0, \quad (3)$$

The problem with employing this strategy in the present case is twofold. First, as we do not have explicit relationship between t and $B(t)$, we can not instantly compute the derivative $B'(t(k))$. Second, the noise to signal² ratio in measuring of both queueing delays and utilization can be very large (see [22]) especially in the neighborhood of the solution of (2). This would potentially imply low confidence in the estimation of $B'(t)$ in the neighborhood of the solution of (2).

We emphasize again that t is any parameter such that by controlling t , one can control both utilization and queueing delays. Thus, if the performance goal is given by keeping the average utilization at a certain level λ , one can design a strategy for achieving that goal by controlling t . Similarly, if the performance objective is keeping the average queueing delay (at the times of congestion) at a prescribed level d_0 , another control strategy can be designed for solving that problem. At this point we should note that by controlling t one can (usually) control not only utilization and queueing delays, but also other (important) performance metrics such as jitter and loss rate. Embedding them into an optimization framework could be done in straightforward manner, but is out of scope of the present paper.

Following the delay-utilization optimization framework developed in the previous section, the performance goal of interest will be the maximization of the benefit $B(t)$. We proceed by presenting an ODU- t algorithm, a strategy with that performance goal.

The ODU- t algorithm controls the variable t such that the value t is updated once per sample time period (Δ) in the following manner:

$$t(k+1) = t(k) \cdot m(k), \quad (4)$$

where $m(k)$ is defined by:

$$m(k) = \alpha, \quad \text{if } \frac{\hat{B}(l(k)) - \hat{B}(l(k-1))}{t(k) - t(k-1)} \geq 0, \quad (5)$$

$$m(k) = \frac{1}{\alpha}, \quad \text{if } \frac{\hat{B}(l(k)) - \hat{B}(l(k-1))}{t(k) - t(k-1)} < 0. \quad (6)$$

Here, $\alpha > 1$ is a constant parameter, close to 1. The choice of α determines the responsiveness of the algorithm. Since t is either multiplied with α or divided by α , in each step k , $t(k) = t(0) \cdot \alpha^{l(k)}$, for some integer $l(k)$. By $\hat{B}(l(k))$ we denote the estimated value of $B(x)$ at the point $x = t(k) = t(0) \cdot \alpha^{l(k)}$. Algorithms of

² By definition $B(t)$ is function of average utilization $u(t)$ and average queueing delay $aQd(t)$. Instantaneous utilization (queueing delay) can be seen as random variable that is sum of $u(t)$ ($aQd(t)$) and appropriate zero mean random variable, that we refer to as noise.

this type can be seen as a version of (3) that do not allow arbitrarily small steps. We again emphasize that strategies of the form of (3) are inappropriate in our problem since any algorithm of type (3) that allows very small changes in the parameter t would suffer from a high noise to signal ratio around global maximum of $B(t)$, and would require a long time for accurate estimation of B in the neighborhood of the global maximum. Moreover, it has been proved in [21], using information-theoretical techniques, that any algorithm for finding an optimum using noisy observations of a benefit function has slow expected convergence. Namely, $O(\epsilon^{-4})$ queries have to be made before one can ensure ϵ -accuracy in the estimation of the optimum x^* . Under dynamic, Internet-like traffic conditions, frequent (small) changes of the traffic patterns might not allow such (exact) algorithms to converge, and can potentially cause undesirable large oscillations.

Algorithms of the form of (4) that do not converge to the certain value, but rather continuously search for the optimal value have been extensively used in the networking literature. Examples of such algorithms are AIMD³ *cwnd* control in TCP [11], AIAD algorithm for controlling the drop probability in BLUE[8] as well as MIMD algorithm for the adaptation of RED parameters in Self-Configuring RED [7].

The parameters of ODU- t are: $P(d)$ - price function, Δ - length of sampling period and α - MIMD parameter. While in general $P(d)$ can be an arbitrary function that satisfies Definition 1, throughout this paper we will mainly use functions that are linear in d :

$$P_\gamma(d) = \gamma d, \quad \gamma > 0. \quad (7)$$

If we restrict ourselves to price functions of this form then the parameter $P_\gamma(d)$ can be specified by a single scalar γ . A higher value of γ assigns more importance to low delays and vice versa. The sampling period time Δ should be chosen to cover several “typical” round trip times, in order to allow traffic to respond to change of t . Choosing Δ in range [1sec, 5sec] usually satisfies this condition. The parameter α determines the responsiveness of ODU- t , and should be selected such that it allows doubling/halving of t within several seconds (up to one minute).

At this point we discuss the notion of variability in the traffic conditions. Measurements from [20] show that on typical 150Mbps+ links, basic IP parameters such as the number of active connections, the proportion of TCP traffic, the aggregate IP traffic, etc., do not change dramatically. Although we do not exclude the possibility that there can be drastic changes in the traffic mixes, our basic presumption in the design of ODU- t is that such events are rare enough to be considered as exception rather than rule. Thus, ODU- t is designed to search for an optimal solution in the “regular” intervals, during which traffic conditions vary slowly. In the cases of dynamic traffic conditions, one can perform self tuning of the parameters Δ and α depending on the level of changes in the traffic conditions.

³ Additive Increase Multiplicative Decrease.

The following theorem shows that, assuming that estimator \hat{B} preserves order of B on the grid $\mathcal{T}_\alpha = \{t(0) \cdot \alpha^n, n \in \mathbb{Z}\}$ the controller (4) runs system to the state that is close to global optima. The proof is given in [22].

Theorem 1. *Let t^* be the point where global maximum of B is attained. Suppose that estimator \hat{B} preserves the order on the grid \mathcal{T}_α , ie. for all $m_1, m_2 \in \mathbb{Z}$:*

$$\hat{B}(m_1) \geq \hat{B}(m_2) \Leftrightarrow B(t(0)\alpha^{m_1}) \geq B(t(0)\alpha^{m_2}).$$

Then there exist m_0 such that for all positive integers r :

$$t(m_0 + 2r) = t(m_0 + 2r + 2) = \bar{t}$$

$$t(m_0 + 4r + 1) = \bar{t}\alpha, \quad \text{and} \quad t(m_0 + 4r - 1) = \frac{\bar{t}}{\alpha},$$

and the relative error between \bar{t} and t^* satisfies:

$$\frac{\bar{t} - t^*}{t^*} \leq \alpha - 1. \blacksquare$$

4 Case study: t is the available DropTail space

In this section we evaluate the behavior of ODU- t for t denoting the available DropTail space.

Simulation⁴ 1. Our first set of simulations illustrate the dynamics of t under static conditions of 50 TCP flows with RTT's uniformly distributed in range $[20, 220]msec$ and with packet sizes of 1000 bytes. We run ODU- t with parameters $\Delta = 2sec$, $\alpha = 1.05$. The price function used is $P_{10}(d) = 10 \cdot d$. Initially: $t(0) = 100Kbytes$. The off-line (see Simulation 2) optimal value are approximately $t^* \approx 130Kbytes$. Figure 1 depicts the queue occupancy, evolution of t and utilization for both cases.

Simulation 2. The second set of simulations shows how close the average queueing delays and average utilization are to the optimal values, in static conditions with a constant number TCP flows. We ran the set of 50 TCP flows, with RTT's uniformly distributed in range $[20, 220]ms$ and packet sizes of 1000 bytes, over a bottleneck link with capacity 10MBps. By running a sequence of simulations with buffer of *constant* size we can empirically find $aQd(t)$ and $u(t)$ and thus the optimal values t^* corresponding to different price functions. We refer to these (empirically obtained) optimal values as offline-optimal. Rows 3,5 and 7 in Table 2 contains aQd , u and B_γ for the offline-optimal value of parameter that maximize B_γ in three cases: $\gamma = 2, 10, 20$. For same value of γ 's we run ODU using price function $P_\gamma(d) = \gamma \cdot d$ as the parameter. Online averages (5 minutes per simulation) of aQd , u and B_γ are presented in Table 2 in each of these three cases.

⁴ Scripts used in all simulations from this paper can be found at <http://www.hamilton.ie/person/rade/Optimal/>

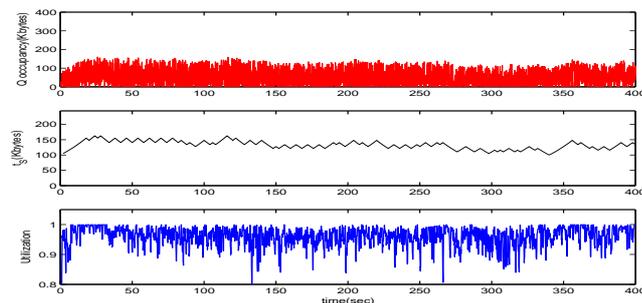


Fig. 1. Simulation 1. Queue occupancy, available buffer space(t), and utilization for ODU- t queue servicing 50 TCP flows.

Scheme, γ	$aQd(sec)$	u	B_γ
ODU, $\gamma = 2$, online	0.01075	0.9890	0.9675
DT, $\gamma = 2$, off-line	0.01058	0.9876	0.9664
ODU, $\gamma = 10$, online	0.00471	0.9544	0.9074
DT, $\gamma = 10$, off-line	0.00521	0.9589	0.9067
ODU, $\gamma = 20$, online	0.00283	0.9222	0.8655
DT, $\gamma = 20$, off-line	0.00293	0.9269	0.8683

Table 2. Numerical results: off-line optima and online ODU- t . The last column represents $B_\gamma(t) = u(t) - \gamma \cdot aQd(t)$.

Simulation 3. This simulation shows stable behavior of ODU in the case of mixtures of TCP and (variable) UDP traffic. In this simulation, the same set of 50 TCP flows that were defined previously compete for a bandwidth on 10Mbyte/sec link, with 50 UDP flows that have exponentially distributed on and off periods. The on-periods have a mean of 1000ms, and the off-periods have mean of 3000ms. The sending rate in on-periods is 1000Kbit/sec. The aggregate UDP arrival rate has a mean of 1.4867Mbyte/sec which is approximately 14.9% of the link's service rate. A histogram, given in Figure 2(left), shows the distribution of the aggregate UDP sending rate sampled on 100ms intervals.

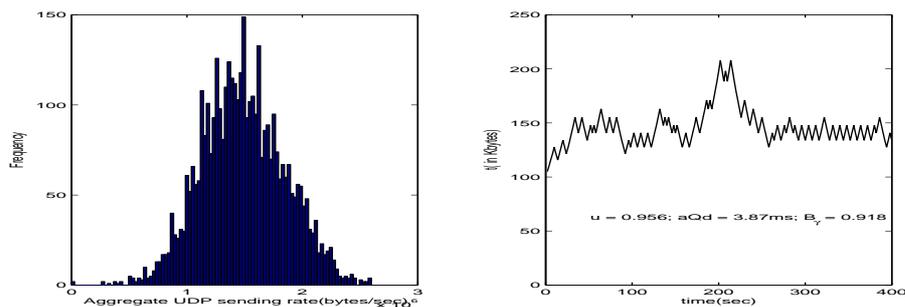


Fig. 2. Simulation 3. Left: histogram of aggregate UDP sending rate (sampling intervals 100ms). Right: available buffer space as function of time. Queue service 50 TCP flows and 50 on-off UDP flows.

The ODU parameters are the same as in previous simulations: $\Delta = 2sec$, $\alpha = 1.05$. The price function used in both cases is $P_{10}(d) = 10 \cdot d$. Initially: $t(0) = 100Kbytes$. Figure 2(right) depicts evolution of t together with obtained values of average utilization, average queueing delay and benefit .

We refer the reader to [22] to explore the behavior of ODU- t some other scenarios which: empirically show stable behavior in cases of sudden changes of traffic pattern; compare ODU that controls DropTail queue size with ODU that controls per-packet drop probability; etc.

Simulation 4. Here we demonstrate how other performance metrics are impacted by changes in available DropTail buffer space. We concentrate on fairness and loss rate. We use Jain's Fairness Index (JFI) [12] as a fairness indicator and is defined as follows. For set of users u_1, \dots, u_k let $r = (r_1, \dots, r_k)$ be vector of their achieved average rates during the measurement interval. Then

$$JFI(r) = \frac{\left(\sum_{i=1}^N r_i\right)^2}{N \sum_{i=1}^N r_i^2}. \quad (8)$$

The simulation setup is same as in Simulation 1 and consists of 50 TCP flows serviced by the 10MBps bottleneck link with RTT's uniformly distributed in $[20, 200]ms$. The bottleneck link has a DropTail queue with size of S kilobytes. We varied S in range 10 to 300. A basic observation is that the performance of TCP-like congestion control algorithms, whose dynamics depend on round-trip time, is significantly affected by queueing delays. By increasing the queueing delay, the aggressiveness of TCP senders is decreased, implying lower loss rates. From a fairness perspective, larger queueing delays decrease bias against long-RTT connections. Indeed, for two TCP connections, with round trip times RTT_1, RTT_2 , $RTT_1 < RTT_2$, bottlenecked at a single link with queueing delay d_0 , the ratio of their expected rates⁵ is $\frac{RTT_1 + d_0}{RTT_2 + d_0}$. Increasing d_0 leads this ratio to a value closer to one. Figure 3 presents the dependance between available space in FIFO Drop-Tail queue and loss rate and JFI. We note that for very small queue sizes (< 50 kilobytes), loss rates are large and TCP dynamics is dominated by timeouts. In this regime the square root formula is not valid and fairness is impacted mainly by timeout mechanism.

5 Summary

In this paper we have addressed the problem of utilizing the tradeoff between queueing delays and link utilization. By specifying the relative importance of queueing delays and utilization, an optimal choice of a queue management parameter is the one that maximizes the overall benefit defined by (1). There could be two possible approaches for solving this problem. First, suppose that one can, by accurate modelling and effective estimation, predict the delay/utilization dependance from the control parameter. Then, by an off-line solving of the underlying optimization problem we can set the parameter that controls queue scheme

⁵ This follows from the square root formula.

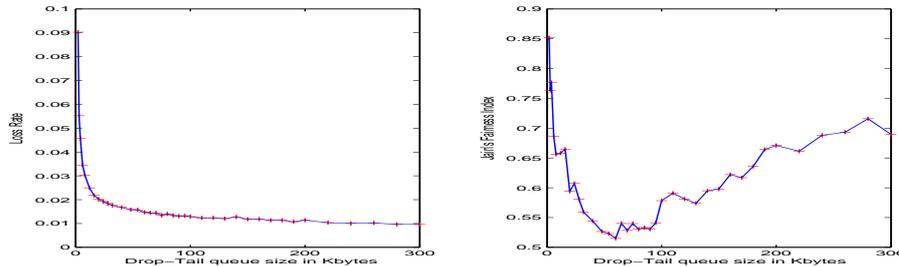


Fig. 3. Simulation F. Loss rates (top) and JFI (bottom) for 50 TCP flows serviced by Drop-Tail queues of different sizes.

to the optimal value; see [2] for one strategy of this type. And second, where one can adapt the control parameter such that on average the overall benefit is maximized. We argue, that the first approach is not feasible in the current Internet because of both nonexistence of accurate and tractable enough models for the delay/utilization dependance, and the highly nontrivial estimation techniques that such an approach would require. We thus follow the second approach and design an online algorithm Optimal Delay-Utilization control of t which aim to solve the underlying optimization problem by online adaptation of generic parameter t .

The optimization problem (2) assumes a linear dependance between utilization and benefit, and neglected other important performance metrics such as jitter, loss probability, and fairness. In fact, one can define the general overall benefit of the queueing scheme controlled by parameter t as:

$$B_G(t) = V(u(t), aQd(t), j(t), L(t), f(t)), \quad (9)$$

where $j(t)$ is jitter, $L(t)$ is the loss rate, $f(t)$ is a fairness indicator, V is the utility function. We again emphasize the importance of fairness in TCP environments where long-RTT connections could heavily suffer from low queueing delays at the congested links. The embedding of jitter and loss rate into current framework can be done in straightforward manner. However, including fairness into the optimization framework, would be much more challenging as we are not aware of any, computationally light, estimation technique that would faithfully indicate level of fairness. One possible approach to estimate level of the fairness could be by counting runs⁶ as suggested in [16].

From the theoretical point of view, an important open issue is convexity (concavity) of the average utilization/Q-delays/ loss-rates as function of control parameter t (available buffer space, random drop probability, virtual queue capacity, etc). While some results exist for the nonelastic traffic [17, 18], in the case of elastic traffic, the arrival process depends on the control parameter, which makes modelling of the corresponding tradeoff curve more difficult.

⁶ Run is event where arriving packet belongs to the same flow as some, previously arrived packet.

Other AQM schemes could be seen in the optimization framework as well. For example the AVQ algorithm developed in [19] or PI[10] have strict performance goals in terms of utilization(AVQ) or queueing delay(PI). Such schemes can be easily incorporated into our framework, taking appropriate utility functions (see [22]).

The MIMD nature of ODU- t algorithm introduced here is just one possible approach for solving the optimization problem (2). In Section 3 we discussed the rationale for choosing MIMD algorithm that continuously searches for optimal value instead of an algorithm that will search for an exact optimal value under noisy measurements. It will be interesting to investigate other control strategies as part of future work.

References

1. G. Appenzeller, I. Keslassy, N. McKeown. "Sizing router buffers". ACM SIGCOMM, USA, 2004.
2. K. Avrachenkov, U. Ayesta, A. Piunovskiy. "Optimal choice of the buffer size in the Internet routers". Proc. of the IEEE CDC, Spain, 2005.
3. A. Dhamdhere, C. Dovrolis. "Open Issues in Router Buffer Sizing" ACM CCR, Jan. 2006.
4. A. Dhamdhere, H. Jiang, C. Dovrolis. "Buffer sizing for congested internet links". Proc. of the IEEE INFOCOM, Miami, FL, USA, 2005.
5. M. Enachescu, Y. Ganjali, A. Goel, N. McKeown, T. Roughgarden. "Part III: routers with very small buffers". ACM Computer Communication Review 35(3): 83-90 (2005).
6. C. Estan, G. Varghese, M. Fisk. "Bitmap algorithms for counting active flows on high speed links". Proc. of 3rd ACM SIGCOMM conference on Internet measurement, 2003.
7. W. Feng, D. D. Kandlur, D. Saha, K. G. Shin. "A Self-Configuring RED Gateway". Proc. of INFOCOM, New York, NY, USA, 1999.
8. W. Feng, K.G. Shin, D.D. Kandlur, D. Saha. "The BLUE active queue management algorithms". *IEEE/ACM Transactions on Networking*, vol. 10, no. 4, 513-528, August 2002.
9. R. J. Gibbens, F. P. Kelly. "Distributed Connection Acceptance Control for a Connectionless Network", 16th International Teletraffic Conference, Edinburgh, June 1999, pp. 397-413.
10. C. Hollot, V. Misra, D. Towsley, W.B. Gong. "Analysis and design of controllers for AQM routers supporting TCP flows" *IEEE Transactions on Automatic Control*, pp. 945-959 June, 2002.
11. V. Jacobson. "Congestion avoidance and control". Proc. of the ACM SIGCOMM, 1988.
12. R. Jain. "The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling". John Wiley and Sons, INC., 1991.
13. S. Jaiswal, G. Iannaccone, C. Diot, D. F. Towsley. "Inferring TCP connection characteristics through passive measurements". IEEE INFOCOM, March 2004.
14. H Jiang , C. Dovrolis. "Passive estimation of TCP round-trip times". ACM SIGCOMM Computer Communication Review, v.32(3), July 2002.
15. F.P. Kelly, A.K. Maulloo, D.K.H. Tan. "Rate control for communication networks: shadow prices, proportional fairness and stability". J. Oper. Res. Soc., Vol. 49 (3), March 1998.
16. M. Kodialam, T. V. Lakshman, S. Mohanty. "Runs bAsed Traffic Estimator (RATE): A Simple, Memory Efficient Scheme for Per-Flow Rate Estimation". Proc. of the IEEE INFOCOM, 2004.
17. K. Kumaran, M. Mandjes. "The buffer-bandwidth trade-off curve is convex". *Queueing Systems*, 38 (2001), no. 4, 471-483.
18. K. Kumaran, M. Mandjes, A. Stolyar. "Convexity properties of loss and overflow functions". *Operations Research Letters*, 31(2), 2003.
19. S. Kunniyur, R. Srikant. "Analysis and Design of an Adaptive Virtual Queue (AVQ) Algorithm for Active Queue Management". *IEEE/ACM Transactions on Networking*, 12(2), 2004.
20. Online: <http://pma.nlanr.net/Special/>.
21. B. Pearlmutter. "Bounds on query convergence". Preprint 2005. Online: <http://arxiv.org/abs/cs.LG/0511088>.
22. R. Stanojevic, R. Shorten. "How expensive is link utilization". Technical report, available online: www.hamilton.ie/person/rade/QP.pdf.
23. R. Srikant, *The Mathematics of Internet Congestion Control*. Birkh³auer, 2004.
24. B. Veal, K. Li, D. Lowenthal. "New Methods for Passive Estimation of TCP Round-Trip Times". Proc. of PAM, Boston, MA, USA, 2005.
25. C. Villamizar, C. Song. "High Performance TCP in ANSNET". ACM Computer Communication Review, 24(5), 1994