A robust dynamic solution of the router buffer sizing problem

Christos Yfoulis 1, Christos Xanthopoulos 2

Dept. of Automation
ATEI of Thessaloniki, Greece
1 cyfoulis@autom.teithe.gr 2 xanthopoulos@autom.teithe.com

Abstract—The router buffer sizing problem has been identified recently as an important problem in networking research. Contrary to static proposals and rules of thumb recent efforts have been attempting to develop simple practical algorithms for dynamic queue size management, where the problem is defined on a heuristic optimization basis and the objective is to track a volatile optimum point by continuously tuning at runtime the buffer queue size. In this paper we propose a robust extremum control solution to the problem. Furthermore we evaluate the efficiency of the optimization approach, the practical limitations and the problematic issues of the real implementation. The analysis is based on a plethora of data collected from several representative Ns-2 simulation experiments.

Keywords : router buffer sizing; Ns-2 simulation; QoS; extremum control.

I. INTRODUCTION AND RELATED WORK

The router buffer sizing problem is undoubtedly one of the most popular problems in networking research that has attracted a lot of interest from many research groups and also leaded to a lot of debate in the past decade. The common Bandwidth Delay Product (BDP) rule was the first rule of thumb that was very early adopted (1994) by router manufacturers, but is clearly impractical for today’s and future high-speed computer networks, since it may specify extremely large buffers of high cost and limitations (router design, power consumption, difficulty in employing optical routers). Moreover, another drawback was that the BDP rule was solely based on a 100\% link utilization objective, hence other important metrics, from a user QoS perspective, such as packet queueing delay and loss rate were completely ignored. This rule was challenged by several researchers, who argued that significantly smaller buffer sizes could be used in certain situations, without significant reduction of link utilization. The great advantage of small buffers is a drastic decrease in queuing delays which can have a detrimental effect in real time applications. A recent review of the most important of these efforts can be found in [1].

By adopting link utilization as the only metric of interest, a number of different models have been proposed ranging from the BDP rule ($B = RTT \times C$) to the small-buffer model ($B = RTT \times C/\sqrt{N}$) or $B = (RTT \times C)^2/32N^3$ or the tiny-buffer model (a few dozen KBytes of buffering), where $N$ is the number of connections, $RTT$ the round-trip time and $C$ the capacity of the link. Other constant proposals have also appeared suggesting a buffer of $2L$ packets, where $L$ is the number of input links [2].

However, it was soon realized that link utilization is relatively insensitive to the queue buffer size in many situations and that it is not adequate in terms of the QoS offered, especially for settings where non-TCP traffic, different TCP versions and a variety of different applications with different demands are dealt with. Other important metrics, such as queueing delay and loss rate came into play, and the problem has been transformed to a multi-objective optimization problem, where we seek the minimum buffer size required to keep the link fully utilized while at the same time attempting to keep the loss rate and queuing delay bounded. Other buffer size formulas have been proposed in this context, such as the Buffer Size for Congested Links (BSCL).

The disadvantages of constant size formulas or other formulas expressed as functions of the number of TCP connections $N$ is that they rely on certain binding assumptions, hence they are not generic, and that they assume that $N$ or $RTT$ are known or can be faithfully estimated. However, this is not usually the case, and this leaded several research groups to turn to adaptive buffer sizing (ABS), where the router can adapt its buffer size to its volatile environment (generic Internet TCP and non-TCP traffic consisting of a variety of different applications, bottleneck and non-bottleneck situations, varying and unknown number of connections, link delays and RTTs) by continually updating it at runtime in a closed-loop control fashion. The closed-loop algorithm relies on feedback based on frequent measurements of the metrics of interest (utilization, goodput, queuing delay, loss rate etc.) using flow statistics. Apart from desirable features such as generic applicability and scalability, the fact that high utilization, low loss rate and low queuing delay are conflicting, makes apparent the need of a means for specifying appropriate trade-offs between these metrics, that can be adapted with a flexible manner to specific settings of interest.

Inspired by the ABS ideas, two different types of controllers have been proposed in the literature so far, namely regulation [3], [4] and optimization [5] type controllers. In [3] the authors introduce the active drop-tail algorithm which determines the smallest buffer size that minimizes queuing delay while maintaining a certain predetermined average link utilization. This algorithm is an MIMD-type algorithm (multiplicative increase-multiplicative decrease). Similar ideas are used in [4]...
where an integral controller with adaptive gain is designed such that loss or delay constraints are respected. In [5] a different perspective is proposed. An appropriate cost function capturing the relative importance of different performance metrics (with corresponding weighting factors) is introduced and the buffer sizing problem is transformed to an optimization problem, i.e. maximizing a benefit function that defines desirable trade-offs between utilization, delay, loss rate etc. in an online manner. This is done without any assumptions or models of the traffic conditions, using a MIMD-type of heuristic hill-climbing optimization algorithm.

The focus of this paper is to present a new algorithm for dynamic queue management, which is based on an extremum control heuristic optimization methodology. To this end we adopt a novel hybrid controller that aims at combining the strengths of both constant and adaptive gain controllers, with a view to improving robustness and average performance. These three extremum controllers have been introduced and tested recently in [6], [7] for a different problem, i.e. optimizing the data transfer in queries over Web Services. We also adopt the optimization framework proposed in [5], and study its pros and cons, and several related open issues.

The remainder of this paper is structured as follows. Section II discusses the optimization framework as related to the buffer sizing problem. The extremum controllers are presented in Section III and their evaluation using simulations follows in section IV. Section V concludes the paper.

II. THE OPTIMIZATION FRAMEWORK

According to the optimization framework presented in [5], after including the utilization \( u(q_s) \) and defining appropriate price functions for the average queueing delay \( AvQd(q_s) \) and the loss rate \( Lr(q_s) \) we end up with the total benefit function

\[
B(q_s) = u(q_s) - P_1(AvQd(q_s)) - P_2(Lr(q_s))
\]

where it has been proved, that under static traffic conditions \( B(q_s) \) is a concave function of the queue size \( q_s \), so that the existence of an optimal solution \( q_s^* \) is guaranteed. The simplest choice proposed is to use linear price functions \( P_1(AvQd(t)) = \gamma_1 AvQd(t) \), \( P_2(Lr(t)) = \gamma_2 Lr(t) \). Then, different choices of the tuning “knobs” \( \gamma_1, \gamma_2 \) may specify different relative prices of the queueing delays and/or loss rates, corresponding to different traffic mix scenarios.

To investigate the validity of the above and the need of using a controller, we performed a series of open-loop experiments using Ns-2 packet-level simulations for a representative range of traffic conditions. We consider a typical single-bottleneck topology shown in Figure 1, where a number of sending \( S_i \) and receiving \( D_i \) nodes (users) are included (\( i = 1, \ldots, N \)). The link shared by all connections from router \( R_1 \) to router \( R_2 \) is a bottleneck with propagation delay 50 ms, bitrate of 10 Mbps, and a queue size of \( q_s \) packets. All connections are TCP with round trip times uniformly distributed in range [20,220] ms, packet size 1000 bytes, propagation delay 0.5 ms, and a bitrate of 100 Mbps. All the links employ FIFO drop-tail buffering.

As an initial series of simulations, we conduct experiments for three different values of the total number of connections, i.e. \( N = 10, 50, 100 \) and a varying queue size \( q_s \), ranging from 20 to 300 packets, in steps of 20 packets. We also consider different selections for the weighting factors \( \gamma_1, \gamma_2 \). Our results appear in Figures 2,3,4. The values shown are the averages over 10 runs for fixed values. It must be noticed that, although all graphs appear concave on average, the standard deviation in many cases can be significant enough to insert local optima.

![Fig. 1. Single bottleneck topology used in our experiments.](image)

![Fig. 2. Open-loop simulation experiments with 10 TCP users.](image)

For the particular networking topology, we have selected three values for \( N \) which correspond to different traffic loads, which could be characterized as “Low” (\( N = 10 \)), “Medium” (\( N = 50 \)) and “High” (\( N = 100 \)). These can be representative of the situations encountered when moving from non-bottleneck to bottleneck scenarios. A fully utilized link implies high utilization regardless of the value of \( q_s \), and high queueing delays and loss rates for the whole range of \( q_s \) values. This practically implies that runtime optimization of \( q_s \) is meaningful under non-bottleneck or lightly bottleneck situations, where reducing the buffering to medium (100 – 150 packets) or small values (< 100 packets) could be beneficial in terms of reducing queueing delays of sensitive applications or finding a good trade-off between low delays and loss rates in mixed traffic scenarios.

The figures also reveal that for different selections, there is a variety of concave graphs ranging from monotonically
A specific type of extremum control is examined in this work, namely switching extremum control, and three different flavors are tested, i.e. the constant gain and adaptive gain policies explored in [6] and a novel hybrid technique proposed in [7], that aims at combining the strengths of the constant and adaptive gain policies.

Let $B(t)$ be the performance metric, such as the benefit function and $q_s$ denotes the router buffer size. In switching extremum control, the value of $q_s$ at the $k$th step, $q_s(k)$ is given by the following formula:

$$q_s(k) = q_s(k-1) + g \cdot \text{sign} (\Delta B(k-1) \cdot \Delta q_s(k-1))$$  \hspace{1cm} (2)

where $\Delta u = u_k - u_{k-1}$ and $\text{sign()}$ is the sign function. $g$ corresponds to the gain and can be either constant or adaptive. The formula above can detect the side of the optimum (maximum) point where the current buffer size resides on. The rationale is that the next buffer size must be greater than the previous one, if, in the last step, an increase has led to performance improvement (higher values for the total benefit function), or a decrease has led to performance degradation. Otherwise, the buffer size must become smaller. To mitigate the impact of the noise in the graphs, the measured output and the control input are firstly averaged over a sequence of $n$ measurements.

For an averaging horizon of length $n$, (2) is transformed to

$$q_s(k) = q_s(k-1) + g \cdot \text{sign}(\Delta B(k-1) \cdot \Delta q_s(k-1))$$  \hspace{1cm} (3)

$$\{q_s(k), B(k)\} = \frac{1}{n} \sum_{i=k-n+1}^{k} \{q_s(i), B(i)\}, \ \ k-n+1 \geq 0$$

This controller can be implemented in two ways according to the type of the gain. In the first way, $g = b_1$ is a constant (positive) tuning parameter. In the second way

$$g = b_2 \cdot ||\Delta B(k-1) \cdot \Delta q_s(k-1)||, \ \ b_2 > 0$$  \hspace{1cm} (4)

where $b_2$ is constant. In this case, the step (gain) is adaptive and is proportional to the product of the performance change and the change in the queue size. In [6] both techniques were implemented and the lessons learnt from these efforts has been that there is no clear winner in all cases: adaptive gain may yield more accurate results when the starting point is relatively close to the optimum; policies with constant gain may perform better otherwise, however they exhibit worse behavior during the transient and steady-state phases. The novel hybrid solution proposed in [7] has the form

$$g = \begin{cases} b_1, & \text{in transient phase} \\ b_2 ||\Delta B(k-1) \cdot \Delta q_s(k-1)||, & \text{in steady-state phase} \end{cases}$$  \hspace{1cm} (5)

In the hybrid mode, the step remains constant until the value of $q_s$ converges to a stable value, and then it becomes adaptive. If it re-enters a transient phase, it switches back to constant gain mode. The phase transition criterion proposed in [7] counts the sign switches over a horizon of length $n'$, and defines that a steady-state phase is entered at step $k$ if

$$\sum_{i=k-n'}^{k-1} \text{sign}(\Delta B(i) \cdot \Delta q_s(i)) \leq s$$  \hspace{1cm} (6)
where \( s \) is a small positive integer (odd if \( n' \) is odd, even otherwise). The intuition behind this criterion is that, at steady-state phase, a constant gain switching extremum controller oscillates around the stability point in a saw-tooth manner.

IV. SIMULATION RESULTS

We implemented the three types of controllers and conducted a series of Ns-2 simulations to study their characteristics and evaluate their efficiencies and deficiencies.

Experiment 1: In this experiment we test the effect of different gains for a near-optimal initial decision. Figures 5,6,7 show representative plots of the queue size \( q_s(t) \) returned by the extremum controllers for an initial value \( q_s(0) = 100 \), 50 TCP users and a benefit function \( B(t) = u - d \). In this case the optimal point is located at \( q_s = 120 - 140 \) pks approximately, but it is obvious that the range of benefit function values in the corresponding graph differ only very slightly in the whole range of \([100-180]\) packets (i.e. there exists a relatively wide near-optimal region). We have used three different constant gains \( g = 3, 6, 10 \), three different adaptive gains \( b_i = 5, 10, 20 \) and three different hybrid gains a) \( g=2 \) and \( b_i=5 \), b) \( g=3 \) and \( b_i=15 \), and c) \( g=6 \) and \( b_i=10 \), with \( n = 5, n' = 4 \) and \( s = 2 \). In this particular experiment it is clear that properly tuned constant gain controllers can locate the optimal point. However, mistuned high constant gains may easily lead to a more aggressive and oscillatory response on average that may cause a diversion from the optimal point. Adaptive gain controllers show a more consistent behavior and can yield more accurate results, although again high gains can result in small deviations from the optimal point. Other factors that contribute to this behavior and render fine tuning necessary for accurate response can also be the small range of benefit function values and/or the flatness in the graph.

These observations are partly valid for the hybrid controllers, as well. However, it is clear that a properly tuned hybrid controller \((g=3 \) and \( b_i=15)\) yields the best results, i.e. a fast, accurate and oscillations-free response. Furthermore, hybrid controllers show the most consistent behavior, with smaller deviations from the optimal point when higher or lower gains are used.

Experiment 2: In the second simulation experiment we test the efficiency of the three different controller types for a far from optimal initial decision without relying on finely tuned gains. Figure 8 depicts the queue size \( q_s(t) \). Without taking care for proper tuning of the constant and adaptive gain values, we select \( g = 6, b_i = 10 \). We see that the adaptive gain controller gets stuck before approaching the near-optimal region, while the other two types of controllers eventually manage to approach this region, although their response is unsatisfactory for some time. A closer look to the corresponding benefit function graphs explained the phenomenon. These relatively high-gain controllers were misled by the noise due to volatility and transients (local non-monotonicity) and the small range of benefit function values. However, their ability to continuously search the space allowed both controllers to find their way eventually back to the near-optimal region.

Experiment 3: We now test the extendability and robustness of the controllers to different settings, i.e. when the number of users changes. Figures 9,10 illustrate the performance of the hybrid controller used in the previous experiment \((g = 6, b_i = 10)\) to “High” Load situations with
This paper investigates the problem of optimizing the QoS offered by runtime configuration of the queue size of drop-tail buffers. We presented solutions inspired by extremum control. Using Ns-2 simulations and aggregate measures of important metrics we justified the applicability of a novel hybrid switching extremum controller for locating optimum buffer sizes. However, the lessons learnt from this work is that in a volatile environment, any type of hill-climbing heuristic optimization has to face several practical problems. Apart from noise and local optima, an important issue is the selection of the benefit functions. We have seen that restricted selectivity and/or range of the benefit function values, as well as wide near-optimal regions or poor curvature in the graphs can easily deteriorate the controller’s performance.

There are several paths along which this work could be extended. Other popular flavors of extremum control could be appropriate for the buffer sizing problem, such as e.g. self-tuning extremum control, coupled with on-line system identification. Similar ideas have been already successfully tested in [7]. Finally, as already reported in [10], a new SLA perspective to the buffer sizing problem could be beneficial both in terms of ISP economics and QoS guarantees for their customers. The optimization framework adopted in this work can deal with several performance metrics. Hence, it has the potential to integrate features specified in SLA contracts, such as specified thresholds and related profits or losses, and form the basis on which profit-maximizing feedback controllers could be constructed.

ACKNOWLEDGMENT

Both authors are supported by the ATEI grant titled “Advanced Automatic Control Applications In Computing Systems”.

REFERENCES