

Sizing Internet Router Buffers, Active Queue Management, and the Lur’e Problem

Christopher M. Kellett, Robert N. Shorten, and Douglas J. Leith

Abstract—Recent work in sizing Internet router buffers has shown that for a large number of users, most buffers will be over-provisioned. This work implies that it is possible to significantly reduce queueing delays while incurring little reduction in utilization of the egress link under stationary traffic conditions. While this work is of undoubted value in the context of network research, it is nevertheless difficult to see how these recent results may be used to provision network buffers. In particular, network conditions (traffic mix, number of TCP users etc.) vary over short time scales. Consequently, the bounds derived under the assumption of stationarity make little sense in real network environments. Our principal contribution in this note is to propose an active queue management (AQM) scheme that strives to regulate link utilization by adjusting the drop-tail buffer size. This work exploits recent results on buffer sizing in a way that takes the non-stationarity of the arrival process into account. In particular, our algorithm strives to achieve its target utilization using the smallest possible drop-tail buffer size. In addition, and perhaps more importantly, our formulation reveals a direct connection between the AQM’s stability/convergence and a classical problem in stability theory; namely, the Lur’e problem. Finally, packet-level simulation results are presented to demonstrate the efficacy of our proposed algorithm.

I. INTRODUCTION

Buffers are used at Internet routers to temporarily store incoming packets when the arrival of packets received exceeds the capacity of the egress link. This is done to maintain a high-level of utilization of link capacity and to accommodate bursty traffic. The classical heuristic

Authors are with the Hamilton Institute, National University of Ireland, Maynooth, Co. Kildare, Ireland, and are supported by Science Foundation Ireland Grant 04/IN3/I460. E-mails: chris.kellett@nuim.ie, robert.shorten@nuim.ie, doug.leith@nuim.ie

for sizing router buffers in the Internet, namely the Bandwidth-Delay Product (BDP) rule, states that the amount of buffering necessary to maintain full utilization of a single bottleneck link is given by the product of the bottleneck link capacity C and the ‘typical’ round-trip time RTT between source and destination. However, the BDP rule is becoming increasingly unviable as it implies extremely large network and expensive buffers as link speeds scale up to gigabit level and above.

Recent research has suggested that the assumptions under which the BDP heuristic is useful are overly conservative and several authors have suggested novel sizing strategies that significantly reduce the amount of required buffering in such links. Appenzeller *et al.* [1], and subsequently a number of other researchers, have observed that, under appropriate assumptions, the necessary buffering to maintain high utilization of the egress link is no more than $\frac{C \times RTT}{\sqrt{N}}$, where N is the number of long-lived Transport Control Protocol (TCP) flows accessing the congested router, C is the bottleneck link capacity, and RTT is the round-trip time between source and destination. Avrachenkov *et al.* [2] improved this bound to $\frac{(C \times RTT)^2}{32N^3}$. The underlying rationale for these results is based on the fact that, under typical circumstances, at any congestion event only a fraction of TCP flows reduce their sending rates.

Unfortunately, these sizing strategies are dependent on the number of flows and the mix of traffic in a network, which are dynamic and difficult to estimate quantities. This fact immediately calls into question the utility of such buffer sizing strategies in real communication networks. For example, Figure 1 shows the queue occupancy for

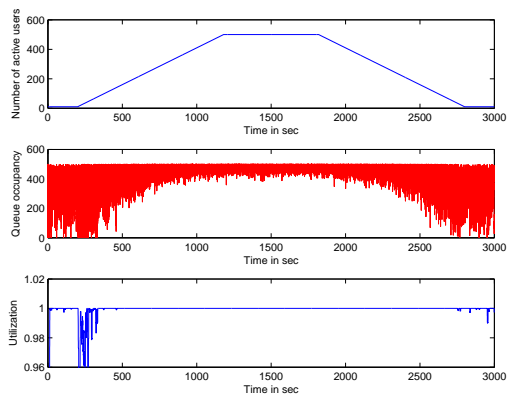


Fig. 1. Slowly varying number of users. Simulation parameters: 200 Mbps link, 500 packet queue, RTTs uniformly distributed between 10ms-300ms.

a single bottleneck link with drop-tail queue as the number of flows is gradually changed. It can be seen that a large persistent queue exists, with only small variations in queue occupancy, when a large number of flows share the link. Hence, the ability of the buffer to accommodate packet bursts is effectively that of a small buffer in the many flow regime and this fact immediately calls into question the need for large buffers in this type of scenario. It can also be seen that when smaller numbers of flows share the link a large queue remains necessary in order to ensure high link utilization. Since the statistics of arriving traffic at a link will typically not be stationary, a buffer size derived under one set of assumptions may be wholly inappropriate in other circumstances.

In addition to non-stationarity, another fundamental problem is that, in practice, we expect link traffic to include a complex mix of bursty, on-off flows, a mix of connection sizes, a mix of TCP and UDP traffic (e.g. voice/video), a mix of congestion control action (not only the many variants of TCP but also TFRC (TCP Friendly Rate Control) and other proprietary streaming algorithms) *etc.* Current approaches to buffer sizing are model-based. That is, a parameterized traffic mix is considered and buffer sizing is analyzed as a function of these traffic parameters (e.g. number of long-lived flows). Such approaches inevitably require consideration of the impact of mismatch between model and reality.

These observations – non-stationarity of traffic arrivals and the need to support complex, poorly characterized arrival processes – lead us naturally to consider an adaptive, measurement-based framework for drop-tail queues. Rather than building ever more elaborate traffic models in an attempt to better capture realistic behavior, we propose a different approach. The quantities of interest: utilization, loss, delay, *etc.* are readily measured for a current choice of buffer size. We argue that this information can be used directly to adapt the buffer size with the aim of regulating utilization. That is, we consider dynamically varying maximum queue length¹ based on feedback of the measured level of utilization, see Figure 2. We refer to this paradigm for active queue management (AQM) as Active Drop-Tail (ADT).

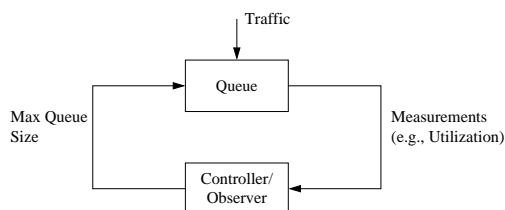


Fig. 2. Active Drop-Tail Architecture.

The ADT paradigm evidently allows the queue to adapt to changing traffic characteristics. Implementation is based on readily available queue measurements (utilization, *etc.*) rather than estimation of arrival process parameters. A key issue in such an adaptive approach is, however, that the traffic arrival process in internet links is typically *elastic*. That is, there exist flow-level feedback loops that seek to roughly match offered load to network capacity. The flow-level feedback loops are coupled together at links where queueing occurs. Changes in the queue parameters can therefore change the arrival process itself. When we adjust the queue parameters in response to changes in the arrival process, this immediately raises concerns as to adverse interactions leading, for example, to oscillations or other instabilities. This stability question is made particularly challenging by the

¹In other words, we regularly compute the maximum allowable queue size and, when a packet arrives, if it causes the buffer to exceed the maximum allowable size, the packet is discarded. Otherwise, it is enqueued.

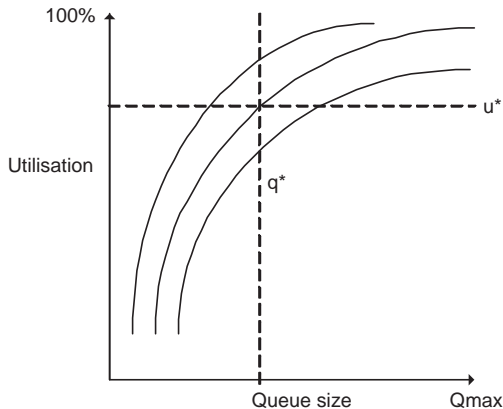


Fig. 3. Utilization curves for a number of traffic scenarios

facts that (i) the arrival process is time-varying, highly complex and poorly characterized and (ii) there is, in fact, a network of queues with queueing possible at multiple bottlenecks simultaneously.

Our contribution in this paper is twofold. Firstly, we demonstrate that the foregoing stability question can be formulated as a Lur'e problem. Consequently we are able account for uncertainty and non-stationarity in network traffic patterns in the form of a sector bounded time-varying nonlinearity. Networks of coupled queues can then be treated inside a passivity or other interconnection framework. Secondly, we illustrate the use of this observation to design an ADT algorithm which strives to minimize queueing delay subject to a minimum desired average target utilization.

II. BUFFER SIZING AND THE LUR'E PROBLEM

We take as our starting point the proposed ADT architecture illustrated in Figure 2. In the sequel we confine consideration to linear control strategies although the approach can be readily extended to encompass nonlinear strategies. Our basic assumption in formulating our stability problem is that average utilization $\bar{u}(k+1)$ is a function of the buffer size $q(k)$ and the traffic arrival process. That is, $\bar{u}(k+1) = f(q(k), \theta_k)$ where θ_k is an exogenous input determined by the traffic arrival process, see Figure 3. We assume that $f(q(k), \theta_k)$ is one-to-one and continuous for fixed θ_k .

The ADT paradigm requires that $\bar{u}(k)$ be estimated at each sample step. This essentially implies

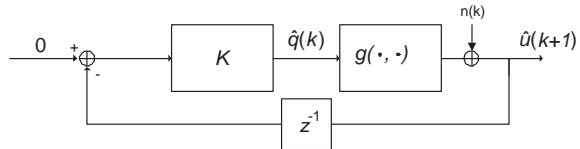


Fig. 4. ADT and the Lur'e problem

a slow sampling rate and the existence of an observer to estimate the average utilization over each sampling period.

Let u^* be the desired reference average utilization of the feedback system, and let q^* be the corresponding equilibrium queue size on one of the possible utilization curves. Define:

$$\hat{u}(k) = \bar{u}(k) - u^*, \quad (1)$$

$$\hat{q}(k) = q(k) - q^*. \quad (2)$$

Then, we can model the relationship between utilization and queue size as

$$\hat{u}(k+1) = g(\hat{q}(k), \theta_k) + n(k) \quad (3)$$

where $g(\cdot, \cdot)$ is a first and third quadrant sector bounded nonlinearity, and $n(k)$ is a bounded exogenous disturbance term due to non-stationarity of the network traffic (i.e. moving between the curves in Figure 3).

Then, the closed loop ADT system can be depicted as in Figure 4 where:

$$k_1 \leq \frac{g(\hat{q}(k), \theta_k)}{\hat{q}(k)} \leq k_2. \quad (4)$$

Furthermore, having restricted attention to a linear controller/observer pair, we see that the ADT paradigm can be seen as a linear system in feedback with a static, memoryless, sector-bounded nonlinearity.

We are interested in the (non-local) uniform asymptotic stability of the equilibrium state of the system depicted in Figure 2. Due to the non-stationary nature of arriving traffic, the disturbance $n(k)$ is typically time-varying and the best we can achieve is BIBO stability. Nevertheless, this is a classic Lur'e problem and can be studied using a plethora of classical tools; most notably the Circle Criterion [7]. A number of remarks are in order in view of the above discussion.

Remark 1: Stability of networks with multiple congested routers can be guaranteed by selecting

controller gains in each router according to the Circle Criterion, and by recalling that networks of connected passive elements are themselves passive. Stability in the large of the entire network follows [11].

Remark 2: The Lur'e/passivity based analysis advocated here can be applied to other AQM schemes; most notably the AVQ framework proposed by Srikant and others [8]. Work will be reported on this topic in future publications.

A. Active Drop Tail (ADT): Convergence and stability

We note that many control algorithms K are possible. As an illustrative example we consider in this paper a simple integrator feedback (see [9] for another control technique):

$$q(k+1) = q(k) + K_{\mathcal{I}}(u^* - \bar{u}(k)) \quad (5)$$

where $\bar{u} \in [0, 1]$ is the utilization estimate provided by the router, $q(k)$ is the available buffering at time $k \in \mathbb{Z}_{\geq 0}$, and K_i is the gain of an integral controller $K = \frac{K_{\mathcal{I}}}{1-z^{-1}}$ (see Figure 4).

In view of the simple controller structure used, BIBO stability follows from the Lur'e formulation using straightforward arguments, provided $K_{\mathcal{I}} \in \left(0, \frac{2}{k_2}\right)$ where k_2 is the sector nonlinearity upper bound.

The preceding stability argument ignores other possible elements in the feedback system such as dynamics of a filter for estimation of $\bar{u}(k)$, saturation elements in the feedback system, and anti-windup elements. However, the effect of such filters/nonlinearities is easily accommodated in the Lur'e framework (see [4] or [7]).

III. SIMULATIONS

We several performed packet-level network simulations using the network simulator *ns-2*². In particular, we looked at queue occupancy and utilization in three scenarios: (i) a step change in the number of users; (ii) slowly-varying number of users (similar to Figure 1); and (iii) a multiple bottleneck network.

²The network simulator *ns-2* is available at http://nslam.isi.edu/nslam/index.php/User_Information. Necessary files for the simulations in this paper are available at <http://www.hamilton.ie/chris/adt>.

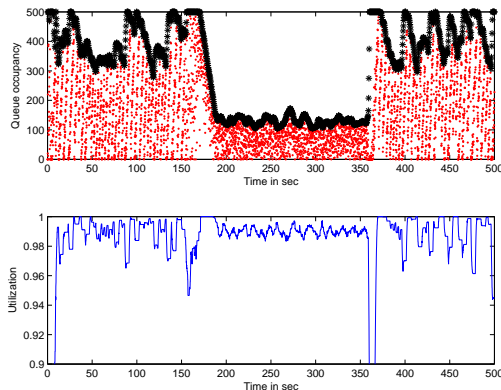


Fig. 5. Step change in users - 20 flows to 100 flows to 20 flows. Simulation parameters: 100 Mbps link, 500 packet queue, RTTs uniformly distributed between 10ms-300ms, controller gain of 750.

A. Step response in number of users

In Figure 5 we show the results of a sudden step change in the number of users. In particular, from time zero we have 20 TCP users accessing a 100Mbps link. At 150 seconds, the number of users increases from 20 to 100. At 350 seconds the number of users returns to 20 flows. We note that the utilization remains near the reference of 99% outside of a short time period when the step change occurs.

Note the persistent, but small, oscillations in the queue size in Figure 5. This is due to the stochastic nature of the arrival traffic as seen in the noise term in (3). However, we do observe the expected BIBO stability.

B. Slowly-varying number of users

By way of comparison with Figure 1, we perform the same simulation with the Drop-Tail queue replaced by an ADT queue, with the results shown in Figure 6. The simulation shows a slowly varying number of TCP users accessing a single 100Mbps link. As the top plot shows, we have ten users up to 200 seconds, after which we add a user every two seconds up to 500 users. From 1800 seconds, a user leaves the network every two seconds down to ten users. Observe that ADT removes the persistent standing queue seen in Figure 1, and thus reduces queuing delay,

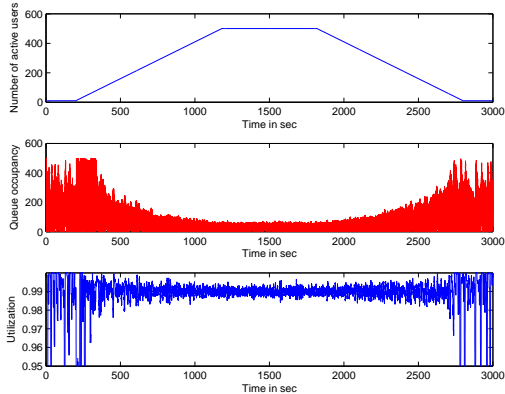


Fig. 6. Slowly varying number of users. Simulation parameters: 300 Mbps link, 500 packet queue, RTTs uniformly distributed between 10ms-300ms, controller gain of 750.

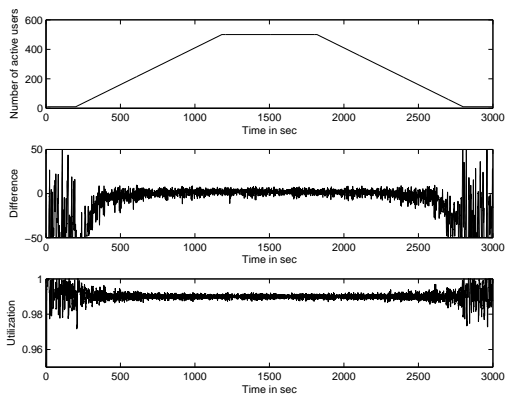


Fig. 7. Slowly varying number of users, comparison with the bound of Appenzeller *et al.* Simulation parameters: 200 Mbps link, 500 packet queue, RTTs uniformly distributed between 20ms-40ms, controller gain of 750.

when more users are present in the network at the expense of a small amount of link utilization.

We perform a similar experiment for comparing our queue limit $q(k)$ with the following bound proposed by Appenzeller *et al.*:

$$B = \frac{C \times RTT}{\sqrt{N}},$$

where C is the link capacity, RTT is the propagation delay, and N is the number of TCP users. In Figure 7, the middle plot shows the difference between the Appenzeller bound and the maximum

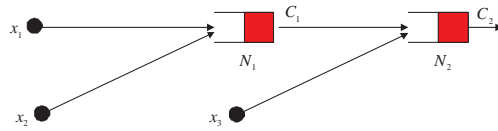


Fig. 8. Multiple bottleneck topology. Simulation parameters: $C_1 = 60$ Mbps, $C_2 = 100$ Mbps, 500 packet queues.

buffer size from ADT; i.e., we plot $\frac{C \times RTT}{\sqrt{N}} - q(k)$, where $q(k)$ comes from (5). Particularly in the regime where the Appenzeller bound is accurate (i.e., in the large number of users regime), the ADT algorithm matches the bound within a few packets either way. Note that, for the purposes of a more accurate comparison with the Appenzeller bound, we have chosen a narrower range of round-trip times.

C. Multiple bottlenecks

Finally, we consider the behavior of ADT in the case where we have two bottleneck links. We consider a two bottleneck aggregation-type topology as shown in Figure 8. We start 50 TCP flows from each of nodes x_1 and x_2 at time zero. At 100 seconds we start another 100 TCP flows from node x_3 . Figure 9 shows that regulation to the target of 99% is achieved for both links. By way of comparison, Figure 10 shows the same scenario with standard drop-tail, rather than ADT, queues. Note that the ADT queues result in significantly reduced queuing delay in both queues, providing a double benefit.

IV. RELATED WORK

Stability analysis of coupled AQM and TCP dynamics has previously been studied using so-called “fluid” approaches (e.g., [5], [8]). We take a different approach and suggest that this approach is preferable to the fluid approach for a number of reasons. Firstly, the insights regarding the buffer size - utilization relationship discussed by Appenzeller *et al.* are valid not only for networks of TCP flows, but in a general sense also for networks carrying complicated mixes of traffic. Fluid models, generally speaking, are unable to capture such complicated network traffic in a meaningful manner. Secondly, the buffer size - utilization relationship exists in a number of situations in

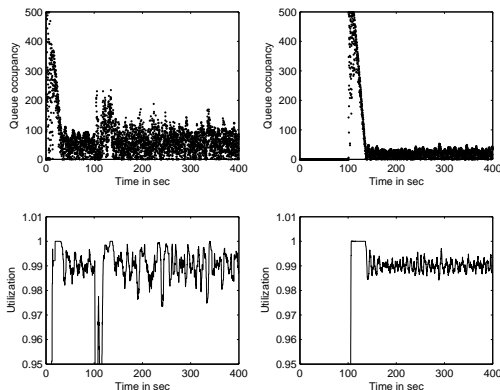


Fig. 9. Left plots: 60 Mbps link; Right plots: 100 Mbps link. RTTs uniformly distributed between 10ms-300ms, controller gains of 750.

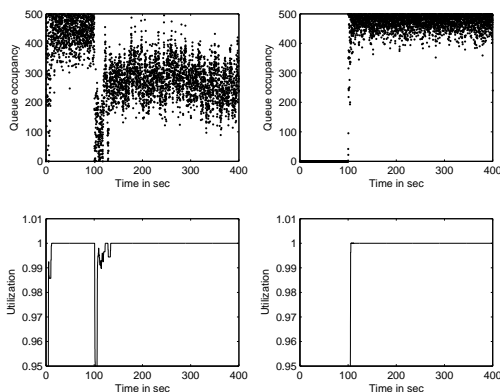


Fig. 10. Left plots: 60 Mbps link; Right plots: 100 Mbps link. RTTs uniformly distributed between 10ms-300ms.

which the use of fluid models cannot be justified; for example, in networks in which low numbers of TCP flows are deployed. Thirdly, the utility of fluid models in the context of drop-tail queues is presently unclear.

As previously noted, ADT is a form of AQM. Many AQM strategies have been proposed over the past fifteen years. The first proposed AQM scheme was RED (Random Early Detection) [3]. RED drops or marks packets according to a probabilistic mechanism based on average queue occupancy in an effort to reduce buffer overflows and improve network fairness. To our knowledge,

the adaptive drop-tail algorithm proposed in [10] was the first algorithm to suggest varying the length of a drop-tail queue. However, rather than reducing queueing delays, the authors' goal was to reduce packet retransmissions and their algorithm will actually increase queueing delays. Adaptive Virtual Queue (AVQ) [6] regulates utilization, similar to ADT, however based on arrival rate, rather than egress link utilization. This means that the mechanism by which AVQ generates drops is different from ADT.

V. CONCLUSIONS

In this paper we consider the Active Drop-Tail (ADT) paradigm for active queue management. ADT is an adaptive, measurement-based framework for drop-tail queues that respects the non-stationarity of traffic arrivals and the need to support complex, poorly characterized arrival processes. A key issue, however, is the stability of the coupled dynamics arising from interaction of ADT queue adjustment and flow-level congestion control feedback. A particular difficulty arises from the fact that the traffic arrival process is time-varying, highly complex and only poorly characterized. We demonstrate that this stability question can be formulated as a Lur'e problem. Consequently we are able account for uncertainty and non-stationarity in network traffic patterns in the form of a sector bounded time-varying non-linearity. Networks of coupled queues can then be treated inside a passivity or other interconnection framework. We illustrate the use of this observation to design an ADT algorithm which strives to minimize queueing delay subject to a minimum desired target utilization.

REFERENCES

- [1] G. Appenzeller, I. Keslassy, and N. McKeown. Sizing router buffers. In *ACM SIGCOMM 2004*, Portland, Oregon, USA, August 2004.
- [2] K. Avrachenkov, U. Ayesta, and A. Piunovskiy. Optimal choice of the buffer size in the Internet routers. In *Proc. 44th IEEE Conference on Decision and Control*, pages 1143–1148, Seville, Spain, 12–15 December 2005.
- [3] S. Floyd and V. Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking*, 1(4):397–413, August 1993.
- [4] G. C. Goodwin, S. F. Graebe, and M. E. Salgado. *Control System Design*. Prentice Hall, 2001.

- [5] C. Hollot, V. Misra, D. Towsley, and W. Gong. Analysis and design of controllers for AQM routers supporting TCP flows. *IEEE Transactions on Automatic Control*, 47(6):945–959, June 2002.
- [6] S. Kunniyur and R. Srikant. Analysis and design of an adaptive virtual queue (AVQ) algorithm for active queue management. In *Proceedings of SIGCOMM'01*, San Diego, California, USA, 27-31 August 2001.
- [7] S. Lefschetz. *Stability of nonlinear control systems*. Academic Press, 1965.
- [8] R. Srikant. *The Mathematics of Internet Congestion Control*. Birkhäuser, 2004.
- [9] R. S. Stanojević, C. M. Kellett, and R. N. Shorten. Adaptive tuning of drop-tail buffers for reducing queueing delays. Submitted to *IEEE Communication Letters*, February 2006.
- [10] J. Sun, K.-T. Ko, G. Chen, S. Chan, and M. Zuckerman. Adaptive drop-tail: A simple and efficient active queue management algorithm for internet flow control. In *Proceedings of the 18th International Teletraffic Congress*, Berlin, Germany, 1-5 September 2003.
- [11] M. Vidysagar. *Nonlinear systems analysis*. Prentice Hall, 1993.