A Self-Tuning Structure for Adaptation in TCP/AQM Networks

Honggang Zhang, Don Towsley
Dept. of Computer Science
University of Massachusetts
Amherst, MA 01003
honggang@cs.umass.edu, towsley@cs.umass.edu

C.V.Hollot
Dept. of Electrical Engineering
University of Massachusetts
Amherst, MA 01003
hollot@ecs.umass.edu

Vishal Misra
Dept. of Computer Science
Columbia University
New York, NY 10027-7003
misra@cs.columbia.edu

Categories and Subject Descriptors

C.2.m [**Computer Systems Organization**]: Computer-Communication Networks-Miscellaneous

General Terms

Design

Keywords

Active Queue Management, TCP, Self-Tuning, Adaptive Control

1. OVERVIEW

Recent advances in the modeling and analysis of TCP/IP networks have led to better understanding of AQM dynamics. Fluid modeling and control theoretic analysis of homogeneous systems [4] has enabled understanding of the relationship between network parameters, such as link capacity C, TCP load N and round-trip time R, to performance objectives, such as AQM responsiveness and robustness. While the analysis dealt with homogeneous systems and long-lived flows, the guiding principles are applicable to realistic network settings with heterogeneous round trip times and short-lived flows. The authors in [4] were able to explicitly tune AQMs such as RED, in terms of these parameters, and also conceive of alternative structures, such as PI [5] in a companion paper, for improved performance. The ability of relating AQM design variables to network parameters opens up the ability to tune the designs using realtime estimates of network conditions. This paper is concerned with such adaptive AQM schemes.

The need for AQM adaptability seems evident; mis-tuned AQMs significantly degrade performance, and network parameters are quite variable. For example, unresponsive traffic such as short-lived TCP flows, or UDP traffic, effectively alter the link capacity experienced by long-lived TCP flows. Also, traditional analysis has assumed a fixed link capacity while considering AQM design. In practice however, a single physical pipe is often divided into several virtual links using various scheduling mechanisms. The capacities of these virtual links are time varying, depending on the scheduling algorithm. Thus, the assumption of a fixed capacity link is often violated in reality. Likewise, dynamic change in the long-lived TCP workload are to be expected, and the TCP/AQM dynamic is quite sensitive to this variation. A conservative design for the "worst-case" load typically leads to degraded performance under normal scenarios.

SIGMETRICS'03, June 10–14, 2003, San Diego, California, USA.
ACM 1-58113-664-1/03/0006.

This work introduces a self-tuning structure (Figure 1) to help AQMs deal with variations in link-capacity and TCP workload. For short we will refer to this class of adaptive AQM schemes as STAQM to emphasize its "add-on" nature; e.g., incorporating it with PI and RED produces STPI and STRED respectively. The scheme has two features: parameter estimation, and AQM tuning.

The concept of parameter estimation is to approximate C, N and R from measurements made locally at the congested router. In this work, we will concentrate on estimating link capacity and TCP load. For round-trip times R we will simply assume that queuing delay is dominated by the propagation delay, a reasonable assumption for current and future high capacity links, and bound the two-way propagation delays based upon physical network considerations. Also, the STAQM will be based on "effective" round-trip times since realistic networks have heterogenous TCP flows. One potential estimation scheme could be based on sampling the SYN and SYN ACK packets of the ongoing TCP flows, but our present focus is on the estimation of C and N.

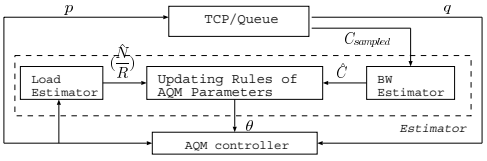

Figure 1: A Self-Tuning AQM Controller Model.

The link capacity C can be estimated by keeping track of departed packets, while an estimate of the TCP load N/R can be inferred from measurements of the (computed) dropping probability p. Indeed, with (C, p) in hand, the TCP throughput equation $\frac{N}{RC} = \sqrt{\frac{p}{2}}$ provides a means for estimating N/R. This is important since the quantity N/R is the *aggregate* incoming throughput, or, router load. By estimating this ratio, we effectively move beyond the analysis in [4], which requires knowledge of the "mythical" number of homogeneous long-lived flows (that is N), to the more physically meaningful effective load N/R. Then, the self-tuning structure is able to automatically tune an AQM based on

these estimates. This requires an explicit parameterization of the AQM in terms of C, N and R. Such parameterizations are available for both RED and PI.

Related work on adaptive AQM includes Adaptive Virtual Queue (AVQ) [6], Predictive AQM (PAQM) [3], Adaptive RED (ARED) [1] and [2], and self-configuring PI [7]. Our self-tuning structure is most closely related to [7]. It differs in that we incorporate estimates of link capacity, provide stability analysis, and gives guidelines of how to choose filter time constants which is crucial to the stability and responsiveness of the STAQM structure. Due to space limitations, these details are relegated to the fuller version of this work found in [8]. Finally, we emphasize that the self-tunig structure presented here is applicable to any AQM scheme that can be parameterized in terms of network parameters C, N and R. The remainder of this paper describes `ns` simulations comparing ARED, PI, STPI and STRED.

2. ILLUSTRATIVE NS SIMULATIONS

Extensive `ns` simulations were conducted in [8] to evaluate the performance of the self-tuning AQMs, STPI and STRED, against fixed PI and ARED. Here we compare the performance of PI, ARED, STPI and STRED when TCP flows experience variations in link capacity C. As previously discussed, such dynamic changes to the TCP workload are to be expected. We consider a rich mix of traffic which contains long-lived TCP flows and short-lived TCP flows (https flows). Figure 2 shows the simulation topology. There are 1000 long-lived TCP flows traveling in both the forward and reverse paths. Round trip delays for TCP flows are uniformly distributed in (40, 250)ms. And, there are 4000 http flows on both the forward and the reverse path.

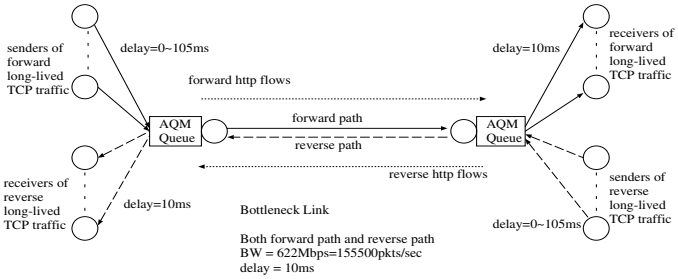

Figure 2: Simulation Topology.

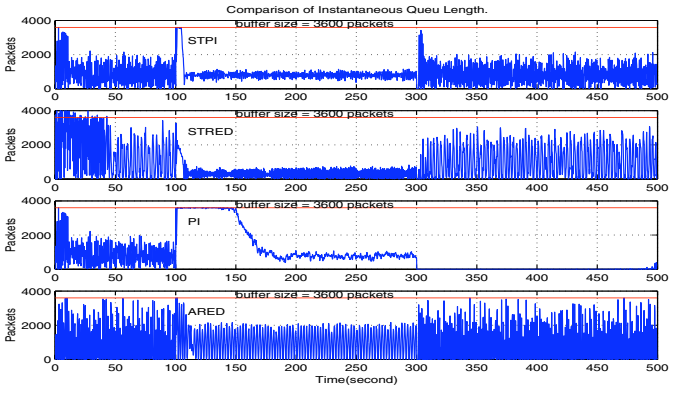

Figure 3: Instantaneous Queue Lengths.

We consider a scenario of varying link capacity in the forward path. Initially, the link capacity experienced by both forward and reverse traffic is 622Mbps. At 100 seconds, we decrease the link capacity of forward path by a factor of 8, then, we return to 622Mbps at 300 seconds. ARED was designed based on the guidelines in [2] to give a target queuing delay of 5ms, and, for comparisons sake, the parameters of STPI, STRED and PI were set to achieve comparable target queue lengths. Specifically, ARED has $min_{th} = 400$ packets. For STPI, The target queue length is 800 packets. For STRED, the target queuing delay is taken as 5ms. [8] gives detailed STRED design guidelines. For both STPI and STRED, the time constants for the estimates of C and $\frac{N}{RC}$ are 5 and 10 seconds respectively. For fixed PI, the target queue length is 800 packets.

The link capacity estimated by STPI is quite accurate, and the traffic load is underestimated as expected. When comparing the accumulative throughputs, we see that STPI has the highest throughput at 500 seconds, approximately 6% higher than that of ARED, and 64% higher than fixed PI. STRED has almost the same throughput as STPI. Figure 3 compares instantaneous queue lengths. STPI regulates the queue length well while ARED and fixed PI oscillate. These oscillations affect queuing delay. Regardless of network conditions, STPI regulates the queuing delay with little jitter. Even though STRED's throughput is almost the same as that of STPI, its queuing delay can *not* be guaranteed.

3. ACKNOWLEDGMENTS

This work is supported in part by DARPA under Contract DOD F30602-00-0554 and by NSF under grants EIA-0080119, ITR-0085848, and ANI-9980552. Any opinions, findings, and conclusions of the authors do not necessarily reflect the views of the National Science Foundation.

4. REFERENCES

[1] W. Feng, D. D. Kandlur, D. Sahar, and K. G. Shin. A self-configuring RED gateway. In *Proceedings of IEEE/INFOCOM*, 1999.

[2] S. Floyd, R. Gummadi, and S. Shenker. Adaptive RED: an algorithm for increasing the robustness of RED. Technical Report, August 2001.

[3] Y. Gao, G. He, and J. C.-J. Hou. On leveraging traffic predictability in active queue management. In *Proceedings of IEEE/INFOCOM*, June 2002.

[4] C. Hollot, V. Misra, D. Towsley, and W.-B. Gong. A control theoretic analysis of RED. In *Proceedings of IEEE/INFOCOM*, April 2001.

[5] C. Hollot, V. Misra, D. Towsley, and W.-B. Gong. On designing improved controllers for AQM routers supporting TCP flows. In *Proceedings of IEEE/INFOCOM*, April 2001.

[6] S. Kunniyur and R. Srikant. Analysis and design of an adaptive virtual queue (AVQ) algorithm for active queue management. In *Proc. of ACM SIGCOMM*, 2001.

[7] W. Wu, Y. Ren, and X. Shan. A self-configuring PI controller for active queue management. In *Asia-Pacific Conference on Communications (APCC)*, Japan, 2001.

[8] H. Zhang, C. Hollot, D. Towsley, and V. Misra. A self-tuning structure for adaption in TCP/AQM networks. Technical Report, University of Massachusetts Amherst, ftp://gaia.cs.umass.edu/pub/Zhang03_staqm.pdf, 2003.