Two Bits are Enough

Ihsan A. Qazi
University of Pittsburgh
Pittsburgh, PA 15260, USA
ihsan@cs.pitt.edu

Lachlan L. H. Andrew
California Institute of Technology
Pasadena, CA 91125, USA
lachlan@caltech.edu

Taieb Znati
University of Pittsburgh
Pittsburgh, PA 15260, USA
znati@cs.pitt.edu

ABSTRACT

We design a congestion control protocol that uses the existing IP ECN bits to achieve efficient and fair bandwidth allocations on high Bandwidth-Delay Product (BDP) paths while maintaining low persistent queue length and negligible packet loss rate. Our protocol uses load factor as a signal of congestion and makes use of a packet marking scheme to obtain high resolution congestion estimates. Our scheme reduces the Average Flow Completion Time (AFCT) by up to 70% when compared with VCP and by up to 45% when compared with TCP SACK+RED/ECN.

Categories and Subject Descriptors: C.2.5 [Computer-Communication Networks]: Local and Wide-Area Networks

General Terms: Algorithms, Design, Experimentation, Performance.

Keywords: Congestion Control, TCP, AQM, ECN.

1. INTRODUCTION

The congestion control algorithm in the Transmission Control Protocol (TCP) has been widely credited for the stability of the Internet. However, future trends in technology, such as increases in link capacities, incorporation of wireless LANs and WANs and proliferation of real-time applications bring about challenges that are likely to become problematic for TCP. In order to overcome the shortcomings of TCP, many congestion control protocols have been proposed. Many such schemes require explicit feedback from the network to aid end-hosts in taking informed decisions. In this category lie schemes such as TCP+RED/ECN, XCP, VCP [5], MLCP [4], RCP [2], etc. However, many of these schemes require more bits than are available in the IP header such as XCP (128 bits), RCP (96 bits) and MLCP (4 bits). Changing the IP header requires a non-trivial and a time-consuming standardization process. Since IPv6 is already being standardized, further changes are unlikely in the near future. Therefore, there is a need to explore techniques for obtaining high resolution congestion estimates using the existing ECN bits. Further, it is important to investigate whether a congestion control protocol based on such schemes can meet the desirable goals of a transport protocol.

As an initial effort towards this end, we design a load factor based congestion control protocol that uses the *Adaptive Deterministic Packet Marking* (ADPM) scheme to obtain congestion estimates with up to 16-bit resolution using the two ECN bits [1]. We use ADPM because it results in the lowest Mean Square Error (MSE) in most cases when compared with REM and RAM [1]. In the context of load factor based congestion control protocols, it was recently shown by Qazi et al. [4] that schemes using two bits for quantization of the feedback signal (such as VCP [5]) have rate of convergence to efficiency that is far from optimal. It was also shown that the feedback must be quantized into at least 16 levels to achieve near-optimal performance in terms of convergence to efficient and fair bandwidth allocations. In this work, we show how this resolution can be achieved using two bits per packet, shared over multiple packets. Further, the use of ECN is done in such a way as to maximize compatibility with RED/ECN.

SIGCOMM'08, August 17–22, 2008, Seattle, Washington, USA.
ACM 978-1-60558-175-0/08/08.

2. FRAMEWORK

In the proposed congestion control framework, each router periodically computes the load factor (ratio of demand to capacity), f, on its output links. This value is then mapped onto the interval $[0, 1]$. We denote the mapped value by c, the congestion level. The mapping from load factor f to congestion level c interpolates linearly between the points $(f, c) = (0, 0)$, $(0.15, 0)$, $(0.75, 0.25)$, $(1, 0.5)$, $(1.2, 1)$, $(\infty, 1)$. In order to obtain high resolution estimates of c, sources employ the ADPM scheme. ADPM leverages the IPid field (used for fragmentation) without changing its functionality. The value in the IPid field, p, is interpreted as a number in $[0, 1]$ by reverse bit counting [1]. When a router with congestion level c forwards a packet, it marks the packet if $c > p$, and leaves the mark unchanged otherwise. The receiver maintains a current estimate of the price, $\tilde{c}$. If it sees a marked packet with $p > \tilde{c}$ or an unmarked packet with $p < \tilde{c}$, then $\tilde{c}$ is set to p. *Intuitively, each arriving packet provides a bound on the value of c at the bottleneck. $\tilde{c}$ is updated whenever a new packet provides a tighter bound on c. As more packets are received $\tilde{c}$ becomes a closer approximation of c.* This estimate is communicated to the sources via the acknowledgements using TCP options. The sources apply Multiplicative Increase (MI), Additive Increase (AI) and Multiplicative Decrease (MD) depending on the congestion level at the bottleneck. The parameters of these control laws depend on the actual value of the price. When $c < 0.25$, the goal of the protocol is *efficiency control.* Sources increase their rates exponentially, proportional to the available bandwidth at the bottleneck. When $c \geq 0.25$, the goal of the protocol is *fairness control* and the sources apply the AIMD control law as illustrated in Figure 1. The MD factor ap-

plied by the sources when $c \geq 0.5$ is a linear function of the amount of overload. This increases responsiveness to congestion and helps improve convergence to fairness.

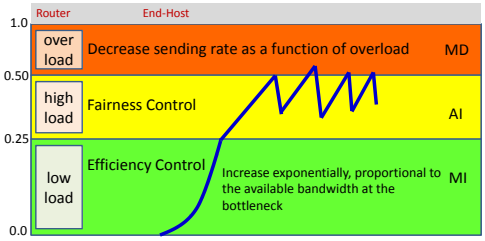

Figure 1: Control laws used by our protocol

3. INITIAL RESULTS AND INSIGHTS

We have developed models to determine the probability distribution of the price estimation error. Of particular interest is the time to detect overload (*i.e.*, when $c \geq 0.5$). Using a Markov Chain model, we find that in steady-state, flows detect overload in the first RTT after congestion with high probability. This insight leads us to approximate the probability of congestion detection with a geometric distribution. In particular, the probability that flow i detects congestion in the first round after overload is well approximated by

$$1 - (1 - p_d)^{w_i(t)} \quad (1)$$

where p_d=$c - 0.5$ is the probability that a given packet receives a mark and $w_i(t)$ is the window size of flow i. The above expression implies that low rate flows will detect overload later than high rate flows. This is a desirable consequence of ADPM because it helps convergence to fairness. As long as high rate flows react to congestion, they allow low rate flows to quickly reach the equilibrium window sizes without causing overload at the bottleneck. p_d increases roughly by $N\alpha/[k(u-1)]$ every RTT after overload, where N is the number of flows, α is the AI parameter, k is the BDP of the path and u>1 is a variable that controls the aggressiveness of response. We find that the probability of detecting overload remains roughly invariant to the BDP of the path. The reason is that on high BDP paths, flows either have large windows (when N is small) or high N which gives rise to higher p_d. To see this more clearly, note that $1-e^{-p_d w_i(t)}$ provides a good approximation of Eq.1 when p is small and $w_i(t)$ is large. In steady-state (when flows have achieved their fair rates) $w_i(t) \approx (k+N\alpha)/N$ in overload, the expression becomes $1-e^{-N\alpha(k+N\alpha)/Nk(u-1)}$. If $k/N \gg \alpha$, it roughly equals the constant value $1-e^{-\alpha/(u-1)}$ (see [3] for details of the results).

Figure 2 shows the improvement in AFCT that our scheme brings over other schemes as a function of the average file size using ns2 simulations. Let r_s be the AFCT of scheme s and r_p the AFCT of the proposed scheme. The improvement is expressed as $(1-r_p/r_s)100\%$. *Note that our scheme offers a reduction in AFCT of at least 40% and up to 70% over VCP (the 2-bit counterpart of our scheme that does not use ADPM) and up to 45% over SACK+RED/ECN and up to 38% over MLCP* [4]. For these experiments, we considered a single bottleneck topology with a bottleneck capacity of 10Mbps and an average RTT of 200ms. The file sizes

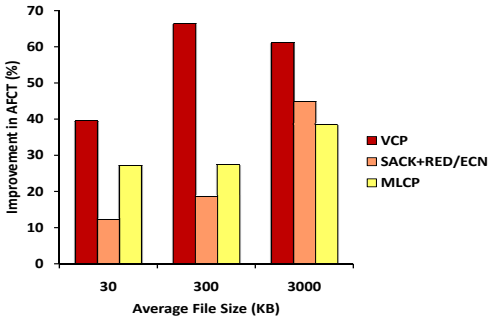

Figure 2: The improvement in AFCT that our scheme brings over other schemes as a function of the average file size.

obey the Pareto distribution with a shape parameter of 1.2. The average file size was varied from 30KB to 3MB and the offered load was kept at 0.9. Note that as the average file size increases, our scheme provides a greater improvement over SACK+RED/ECN. The reason is that when the average file size is small, most flows remain in slow-start during their lifetime. However, as the average size increases, flows have to enter the slow congestion avoidance phase which increases their duration.

4. CONTRIBUTIONS AND FUTURE WORK

Our contributions can be summarized as follows:

- We show that it is feasible to use the existing ECN bits to convey congestion price estimates of as high as 16-bit resolution without sacrificing performance due to estimation errors.
- We design a load factor based congestion control scheme that uses ADPM for packet-marking at the routers. Our scheme closely approximates the performance of an optimal load factor based scheme.
- We develop analytic models and conduct extensive ns2 simulations to characterize the performance of our scheme. Our analysis provides novel insights into the design of load factor based congestion control protocols. These insights are likely to lead to better designs for next-generation congestion control protocols

In load factor based schemes, rate of convergence to fairness is slower than in schemes such as RCP. As part of our ongoing work, we are investigating ways to improve convergence. Further, we are in the process of doing the stability analysis of our protocol.

5. REFERENCES

[1] L. L. H. Andrew, S. V. Hanly, S. Chan, and T. Cui. Adaptive deterministic packet marking. *IEEE Comm. Letters*, 10(11):790–792, Nov. 2006.

[2] N. Dukkipati, M. Kobayashi, R. Zhang-Shen, and N. McKeown. Procesor Sharing Flows in the Internet. In *IWQoS*, Jun 2005.

[3] I. A. Qazi, L. L. H. Andrew, and T. Znati. Two bits are enough. Technical report, University of Pittsburgh, April 2008. URL: `http://www.cs.pitt.edu/~ihsan/twobitsenough.pdf`.

[4] I. A. Qazi and T. Znati. On the Design of Load Factor based Congestion Control Protocols for Next-Generation Networks. In *IEEE INFOCOM*, Apr 2008.

[5] Y. Xia, L. Subramanian, I. Stoica, and S. Kalyanaraman. One More Bit Is Enough. In *ACM SIGCOMM*, Aug 2005.