The Impact of False Sharing on Shared Congestion Management *

Aditya Akella† Srinivasan Seshan†

1 Introduction

Recently, several proposals have been made for sharing congestion information across concurrent flows between end-systems. In these proposals, the granularity for sharing has ranged from all flows to a common host to all hosts on a shared LAN. While these proposals have been successful at ensuring sound AIMD behavior of the aggregate of flows, they suffer from what we term "false sharing". False sharing occurs when two or more flows sharing congestion state may, in fact, not share the same bottleneck.

In this work, we investigate the effects of false sharing on shared congestion management schemes. We characterize the origins of false sharing into two distinct cases: (i) networks with QoS enhancements such as differentiated services, where a flow classifier segregates flows into different queues, and (ii) networks with path diversity where different flows to the same destination address are routed differently – a situation that occurs in dispersity routing, load-balancing, and with network address translators (NATs). We evaluate the impact of false sharing on flow performance and consider whether it might cause a bottleneck link to become persistently overloaded. We propose schemes for detecting false sharing and show how different metrics (loss rate, delay distribution, and reordering) compare for this purpose. We also consider the issue of how a sender should respond when it detects false sharing.

2 Results

One danger of false sharing is that the slower of two senders may be confused by the faster sender and, as a result, may send at a rate faster than its bottleneck link can sustain. Our simulation results show that even under extreme false sharing conditions, window-based congestion control algorithms force the faster sender to send at a slower rate and thus ensure that the slower sender does not overload any links. Figure 1 illustrates the simulation results of such a false sharing situation. This simulation tests 10 flows belonging to the DiffServ Assured Forwarding (AF) class and 40 Best Effort (BE) flows under different bandwidth allocations to the AF and BE classes. The figure plots the bandwidth achieved by a single AF and BE flow when congestion sharing is performed and when it is not. The bandwidth achieved under congestion sharing is clearly lower than either flow under any conditions - ensuring that sharing never leads to overload. We have also analytically modeled this behavior. The plot of this analytic throughput prediction closely matches the simulation results in Figure 1.

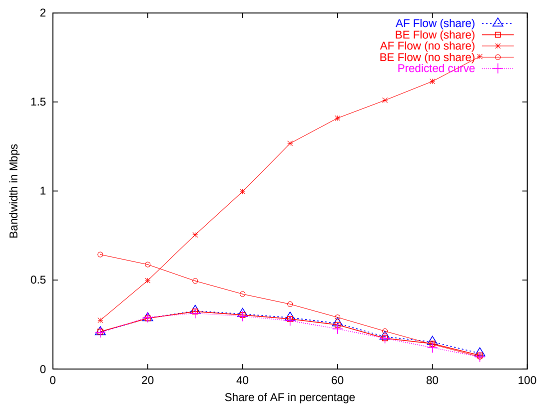

Figure 1: The impact of false sharing in a Service Differentiation setting. The curves for AF and BE flows (which share congestion information) and the analytical prediction overlap.

We have also developed tests to detect false sharing based on delay and loss correlation between packets across the participating flows. Our tests show that delay and reordering statistics are superior to those based on loss patterns in terms of detection time and accuracy. We have also found that for such tests, it is markedly easier and quicker to identify when false sharing is occurring than to identify situations where congestion information can be shared safely. For example, in our simulations, the tests could typically identify false sharing in 5-10 seconds but could only identify "safe" sharing in 60-70 seconds.

Based on these observations, we believe that the default behavior of congestion sharing systems should be to share congestion state aggressively and react to false sharing. This design is motivated by three key observations: 1) false sharing never results in dangerous congestion control behavior or network overload, 2) it is easier to detect false sharing than safe sharing, and 3) the detection tests work best when packets from different streams are sent in an interleaved fashion and such scheduling only occurs when sharing is performed. Using this design, the key concern becomes the performance degradation of the faster flow. Our simulations show that the performance of the faster sender recovers completely in approximately 3 times the period taken to detect and react to false sharing.

3 Future Work

Most of our current results are based on simulations of relatively long-lived flows. We are confident that our schemes would work well even with short flows given the robustness of the metrics. We plan to implement these algorithms in the Linux kernel and test out the validity of our hypotheses in the wide-area. We expect that these real-world tests will require us to face the challenges of measurement-noise and realistic traffic patterns.

*A technical report that describes this work is available at http://www.cs.cmu.edu/∼aditya/papers/tech-rep.ps

†School of Computer Science, Carnegie Mellon University, Pittsburgh, PA. {aditya+, srini+}@cs.cmu.edu