

Reflections on the TCP Macroscopic Model

Matthew Mathis
Pittsburgh Supercomputing Center
mathis@psc.edu

This article is an editorial note submitted to CCR. It has NOT been peer reviewed.

The author takes full responsibility for this article's technical content.

Comments can be posted through CCR Online.

ABSTRACT

The current Internet fairness paradigm mandates that all protocols have equivalent response to packet loss and other congestion signals, allowing relatively simple network devices to attain a weak form of fairness by sending uniform signals to all flows. Our paper[1], which recently received the ACM SIGCOMM Test of Time Award, modeled the reference Additive-Increase-Multiplicative-Decrease algorithm used by TCP. However, in many parts of the Internet ISPs are choosing to explicitly control customer traffic, because the traditional paradigm does not sufficiently enforce fairness in a number of increasingly common situations. This editorial note takes the position we should embrace this paradigm shift, which will eventually move the responsibility for capacity allocation from the end-systems to the network itself. This paradigm shift might eventually eliminate the requirement that all protocols be "TCP-Friendly".

1. INTRODUCTION

My co-authors and I are very honored to receive the Test of Time Award for our paper describing the $1/\sqrt{p}$ model of TCP performance[1]. Very few people receive this award and we are grateful for this once-in-a-lifetime honor. However, I would like to use this opportunity to reflect on the underlying assumptions from which the model was derived and consider if these are still the best choices for the entire Internet.

Our paper survived the test of time due to the continued acceptance of the established Internet fairness paradigm: shared bottlenecks sending the same congestion signals (primarily packet loss) to all flows, which are mandated to have equivalent responses to these signals. It modeled the reference Additive-Increase-Multiplicative-Decrease (AIMD) congestion control algorithm and became the basis for TCP-friendly rate control[2]. The model helped to cement the idea that Internet fairness could be implemented with simple network devices and standardized end-system behavior[3].

However, I argue that it is time for the Internet to move beyond this paradigm, by progressively shifting more of the responsibility for allocating capacity from the end-systems to the network itself. This idea is not new[4, 5], but has been getting progressively more traction due to a number of fairness problems, including:

- Non-responsive, non-standard protocols, typically based on UDP.
- Peer-to-peer and other applications that open large numbers of connections.
- The egregious unfairness of standard TCP when very short RTT flows compete with wide area flows. This can be a major problem in a number of settings such as data centers and university campuses. The symptoms of this problem will become more pronounced as TCP autotuning continues to be rolled out in Vista, Linux, Mac OS, and various BSD unixes.
- The increasing deployment of autotuning also magnifies the problems associated with drop-tail queues[6], since every single TCP flow with sufficient data has the potential to cause congestion somewhere in the network.

Due to these fairness problems, most ISPs are finding that they need to protect paying customers from others who don't share so well. Generally they do so by throttling traffic at or near the access links facing their customers. In some cases the throttling is as simple as only offering their customers limited bandwidth. In other cases ISPs implement some form of Fair Queuing, or sometimes other more complicated fairness enforcement mechanisms. The key feature is that the network provides strong isolation between flows, allowing it to send strong feedback signals to greedy flows without disturbing smaller flows.

ISPs are generally careful to build their networks such that their core and interconnections with other ISPs are uncongested. If they don't, their largest customers are likely to complain about poor or erratic performance. The ISPs can do this because their costs are dominated by their customer access links. It only takes a relatively small share of the revenue from each customer to maintain reserve capacity elsewhere in the network. In this manner, the core can be protected from congestion by pervasive bottlenecks at or near

the access links. These are the very same bottlenecks that are being used to throttle greedy customers.

2. A STRONGER PARADIGM

As the network evolves to exert more control over the traffic, the notion of “TCP-friendly” will become increasingly irrelevant. As more non-TCP-friendly congestion control algorithms are deployed, they in turn create additional pressure on the ISPs to exert more or better control over the traffic. With a little help from the SIGCOMM community it might be possible to trigger an Internet wide phase transition to a different Internet congestion control paradigm.

In this paradigm, TCP’s role would be to efficiently maintain queues at network bottlenecks. The network’s role would be to allocate capacity at the bottlenecks according to some policy, which is likely to preferentially allocate capacity to real time and small flows before larger flows. Note that the capacity allocation policy should reflect any business conversation between the ISP and its customers (e.g. marketing claims). These policies do not need to fit any particular abstract notion of fairness nor do they need to be standardized across providers.

Since the network can send independent congestion signals to each flow, it can enforce its own capacity allocation policy even if each flow has a different response to congestion signals. Congestion control would not need to be standardized, except to meet some yet to be defined Internet safety and stability requirements, primarily avoiding congestion collapse and not wasting capacity. This paradigm shift would open the doors for context-specific TCP implementations designed for adverse environments, such as lossy wireless links, long fat networks or even long fat lossy networks.

Many people have discovered alternate congestion control algorithms with interesting and useful properties, except they do not share appropriately in the presence of uncontrolled network bottlenecks. By and large these algorithms have been discarded without ever being formally evaluated or published as candidates for deployment in the Internet. These previously ignored algorithms might be perfectly acceptable under this proposed paradigm.

Needless to say our time tested model will no longer generally apply. C’est la vie.

3. EVOLVING THE INTERNET

Existing market forces are already pushing the Internet along the path to this paradigm. As I noted earlier, many ISPs are explicitly managing their traffic while their customers, knowingly or otherwise, are striving to get their “unlimited fair share” of the network ahead of other greedy users. If, as a community, we embrace and guide the deployment of non-AIMD-friendly¹ protocols, we will eventually arrive at an Internet that has a greatly enriched spectrum of acceptable

¹Note that although the term “TCP-friendly” reflects the history of the prevailing congestion control paradigm, it isn’t really accurate and clashes badly in some usages. Would you describe a non-AIMD TCP as being “not TCP-friendly TCP”? AIMD-friendly captures the original spirit and does not suffer from such cognitive dissonances.

behaviors for both network devices and congestion control algorithms. As long as the evolution in congestion control algorithms does not happen too quickly, or in uncontrolled ways, the ISPs and equipment vendors will be able to make sufficient changes to the networks quickly enough to protect their customers from each other.

This is not to say that the transitions will be painless or easy, or that there is not a lot of work to be done on the way. It will have to be a very gradual evolution, spanning many years. There are three broad areas that need attention – evolving the network, evolving congestion control and managing the transition. Some open challenges include:

- The requirement that protocols be AIMD-friendly has largely supplanted the need for a good theoretical understanding of the nature and causes of congestion collapse and other pathologies. There have been some important inroads into understanding generalized end-to-end congestion control in terms of stability[7] and tendency to cause congestion collapse[8], but these studies abstract away many details that may be profoundly important under crisis conditions. We do not have a general understanding of, nor tests for, protocol features that might be prone to congestion collapse or other types of undesirable behavior. We need to replace the “TCP-Friendly” test with some other test to decide if new non-AIMD-friendly algorithms or protocols are safe for the Internet.
- When there are observed fairness problems, consider carefully whether they should be considered end-to-end congestion control problems or network problems. Be sure to blame the network when the network deserves it. For example, if there are drop tail queues without Active Queue Management, be wary of problems caused by full queues, such as lock-out[6]. Potentially every long queue that is drop-from-tail without AQM should be viewed as a bug waiting to bite someone. Likewise remember that AIMD-friendly is extremely unfair when the RTTs are extremely different.
- Although the edges of the Internet can be protected by some form of traffic control, such as Fair Queuing[4] or Approximate Fair Dropping[5], and the core can be protected by bottlenecks at the edges, there is the potential for there to be some environments which are neither. For example a core link might be congested due to some adjacent failure and have too many flows for FQ or AFD to work well (or to work at all). We need to understand how to mitigate these situations. Additionally are there special problems at university campuses, research centers and other communities that have edge systems with core rate interfaces? Are there simple means to control enough flows in these problematic environments to place bounds on the unfairness? Does Bob Briscoe’s concept of congestion accountability[9] help solve some of these problems?
- What is the right flow granularity and why? What happens when different bottlenecks use different flow granularities? Can congestion accountability be used to improve on past approaches?

In the short term we want to relax the absolute requirement that all protocols be AIMD-friendly all the time. The first steps are to encourage the IETF to formally sanction the use of non-AIMD-friendly congestion control algorithms in some well defined, controlled environments, and second, to become progressively more lenient about algorithms that are only approximately AIMD-friendly. To some extent both of these changes have already been deployed without sanction. However, until we embrace them as part of natural Internet evolution, we greatly impair our ability to monitor how they actually function in the operational Internet.

4. CONCLUSION

The prevailing paradigm, uniform congestion control algorithms interacting with simple network devices, was absolutely the correct model when the Internet was in its infancy, but we have outgrown it. Today, there is too much at stake to depend on the good graces of everyone to uniformly implement something as critical as congestion control, especially given that our one-size-fits-all stance is no longer a good fit for the huge breadth of technology and scale encompassed by today's global Internet. I suspect that in the not too distant future, it will become clear that the TCP-friendly paradigm was an untenable long term position. We are overdue to move on to a stronger paradigm, and with it, new models for protocol performance.

5. ACKNOWLEDGMENTS

I want to thank all of the authors of the original paper, Jeff Semke, Jamshid Mahdavi and Teunis Ott, whose hard work so many years ago set the stage for the Test of Time Award. Their comments on earlier drafts of this editorial note also helped me to express my views in a way that might be consistent with the sensibilities of the community.

I also want to thank the Test of Time Award selection committee, whom I have to assume did not know that I was already plotting the demise of my own model[10]. They have created an opportunity for me to publicly reflect on it and its underlying assumptions.

And finally I want to thank John Byers, Vern Paxson, Srinivasan Keshav and others at SIGCOMM for encouraging me to write this editorial.

6. REFERENCES

- [1] M. Mathis, J. Semke, J. Mahdavi, and T. Ott. The macroscopic behavior of the TCP congestion avoidance algorithm. *SIGCOMM Comput. Commun. Rev.*, 27(3):67–82, 1997.
- [2] J. Mahdavi and S. Floyd. TCP-friendly unicast rate-based flow control, Jan 1997. Note sent to end2end-interest mailing list.
- [3] Sally Floyd and Kevin Fall. Promoting the use of end-to-end congestion control in the internet. *IEEE/ACM Transactions on Networking*, 7(4), 1999.
- [4] A. Demers, S. Keshav, and S. Shenker. Analysis and simulation of a fair queueing algorithm. *SIGCOMM '89: Symposium proceedings on Communications architectures & protocols*, pages 1–12, 1989.
- [5] R. Pan, L. Breslau, B. Prabhakar, and S. Shenker. Approximate fairness through differential dropping. *SIGCOMM Comput. Commun. Rev.*, 33(2), 2003.
- [6] B. Braden et al. Recommendations on queue management and congestion avoidance in the internet, RFC 2309, April 1998.
- [7] S. H. Low. A duality model of TCP and queue management algorithms. *IEEE/ACM Transactions on Networking*, 11(4), 2003.
- [8] Tom Kelly, Sally Floyd, and Scott Shenker. Patterns of congestion collapse, 2003.
- [9] B. Briscoe, A. Jacquet, T. Moncaster, and A. Smith. Re-ECN: Adding accountability for causing congestion to TCP/IP, July 2008. Work in progress.
- [10] M. Mathis. Heresy following TCP: Train-wreck, April 2008. Note sent to the ICCRG mailing list.