Reflections on "A Control-Theoretic Approach to Flow Control"

S. Keshav
University of Waterloo
Waterloo, ON, Canada
keshav@uwaterloo.ca

This article is an editorial note submitted to CCR. It has NOT been peer reviewed.
The authors take full responsibility for this article's technical content. Comments can be posted through CCR Online.

ABSTRACT

This article discusses the events that led to the publication of my paper 'A Control-Theoretic Approach to Flow Control' that won the Best Student Paper Award in 1991 and a Test-of-Time Award in 2007.

CCS CONCEPTS

• **Networks → Transport protocols**.

KEYWORDS

Reminiscences, Flow Control

As recounted elsewhere in this issue, during 1988 and 1989 I had the good fortune to work with Scott Shenker and Alan Demers at Xerox PARC on Fair Queueing [2]. In the summer of 1989, my advisor at Berkeley, Prof. Domenico Ferrari, arranged for me to do an internship at the legendary Bell Labs, under the supervision of Sam Morgan. That summer, I worked on converting my simulator, REAL [3], from a one-off to production-grade open-source code, incorporating a domain-specific language to specify network configurations and adding support for several networking protocols. At the end of my stay, Sandy Fraser, then Head of Computing Research, generously offered me funding for the rest of my graduate career as well as the opportunity to visit Bell Labs the coming winter, in December 1989.

In Fall 1989, I was working closely with Scott on tuning TCP to make it work well in a network of Fair Queueing routers, but things were not going well. TCP turned out to be extremely sensitive to parameter choice and the slightest bit of tinkering significantly degraded its performance. All Fall, I struggled with it and at the end of November, I had nothing to show for my efforts. Unfortunately, I had to give a talk at Bell Labs on my work in early December, so I was in a state of panic.

I still remember the afternoon of December 1, 1989, when I started work with a fresh research notebook, and I was sitting by the window of a coffee shop on Euclid Street in Berkeley. I was struggling to figure out how an end-system could determine the service rate at an intermediate router. It was clear that if the router queue were non-empty, then a flow could extract pacing information from inter-ack spacing. What was elusive was what to do when the queue was empty. As I was sipping my coffee, the answer came in a flash: to send all packets as back-to-back pairs, so that when they arrived at the bottleneck queue, they would cause a queue. Indeed, if there were many bottlenecks, the worst bottleneck would space them apart the most. It was immediately clear that such a scheme would allow an end-host in a network of Fair Queueing routers to determine the bottleneck service rate with no need for explicit signaling. As I finished my cup, I felt right away that I had hit upon something big, and as things turned out, I was right!

The next week, at Bell Labs, when I gave my talk, the audience loved the idea. I still remember Ellen Hahne, who had done her PhD at MIT on round-robin queueing, telling me how much she loved the idea. So, it was in a state of much excitement that I took time off to visit my parents in Delhi, where I was invited to give a talk at IIT Delhi, my alma mater, in late December. As I was giving my talk, one of the audience members got more and more excited. When the talk ended, Prof. Samar Singh, a faculty member at IIT Delhi, rushed over to tell me that he and his collaborator at the University of Maryland, Prof. Ashok Agrawala, had had the same idea the week before! Indeed, when we went to his office and he showed me their work, it was clear that we had hit upon the same idea. We decided to combine our work and published a joint paper on packet-pair (he called it 2P) in early 1991 [5].

While packet-pair as a tool was effective, it had two significant flaws. First, its estimate of bottleneck service rate, being delayed by up to one RTT, was necessarily stale. Second, it was quite noisy. So, in the first half of 1990, I tried many different approaches to tuning a packet-pair-based flow control protocol to make it robust to delays and noise. My approaches were all based on seat-of-the-pants control algorithms, and none were robust. It looked like my stepping away from TCP was not particularly helpful, after all.

Luckily, I had taken my qualifying exams in late 1989, and at that time, one of my committee members, Prof. Pravin Varaiya, had taken me to task for not learning control theory. He said that if my thesis were to be in congestion control, I ought to be studying this topic! By the summer of 1990, driven to desperation, I decided to learn control theory from the ground up. In Fall 1990, I enrolled in two courses in control theory, one from the Electrical Engineering and Computer Science department, and one from Mechanical Engineering. I found the EECS course to be too difficult to understand, but the ME course was not just easier to understand but also pragmatic. I could see that a solution to my problems was at least feasible, and I started talking to as many control theorists as I could. One of my habits was to write a summary of my work every week and as the months went by, I had more and more notes to work with. It turned out that the problem I was looking at, feedback flow control with delay and noise, was a difficult one. The control theorists I talked to suggested that I look at Linear Quadratic Gaussian

(LQG) control. However, this approach did not work well with noise. Other suggested approaches such as disturbance-accommodation control and state-based control but these also turned out to be dead ends. By late December 1990, I was stuck again.

In January 1991, I decided to take a different tack. I had been taking a class on fuzzy logic with Prof. Zadeh and I thought that fuzzy logic would help in tackling noise. Why not use a fuzzy-logic based Kalman filter, using linguistic rules to adjust the Kalman forgetting factor? This turned out to be quite effective, and, with my classmate Pratap Khedkar, who also happened to be ZadehâĂŹs student, we quickly were able to come up with a very effective fuzzy predictor [6]. That took care of the noise problem, so things were looking promising.

One week before the SIGCOMM 1991 deadline, I thought I had nearly everything in place, but I did not know how to deal with delays. Since I was pressed for time, my strategy was to avoid cutting-edge control approaches and rely instead on classical control. I turned to pole placement, a tried-and-tested approach. With this change, I was able to not only derive a control law but also prove its stability. Success at last! With a few days to spare, I hastily converted my weekly notes and my final theorem into a paper and sent it to SIGCOMM (if you read this paper, you will see what a crude cut-and-paste job it was). I was lucky: despite not having a single numerical result, this paper not only won the Best Student Paper award in 1991, but also a test-of-time award 15 years later. As one of the first papers to combine control theory and flow control in a pragmatic setting, the paper opened the gates for control theorists to work on flow control. I still am proud of this work.

A couple of codas are in order. Once real control theorists entered the area of flow control, I realized that I could not compete with them. So, this was also my last paper on control theory. Nevertheless, I continued to work on a practical implementation of my control law and in 1994, I wrote a long (63-page) paper on "Packet-Pair Flow Control" that contained numerous innovations for the practical use of Packet-pair. Given its length, it could not get published even in a journal, so it was published only as a Bell Labs technical report[1] [4]. The Internet at that time was still dominated by FCFS service, so this paper gathered dust for more than 20 years. Then, in 2018, driven by the fact that Fair Queueing was becoming ubiquitous in data centers, a team the University of Washington implemented Packet-pair and tested it in a data center environment [7], showing that it handily outperformed more recent protocols such as DCTCP [1]. That was sweet vindication!

Perhaps this story has at least some morals for the next generation of researchers. First, the hoary advice to "follow your passion" is really true. My work was immensely frustrating and what kept me going was my dream of a flow control protocol that worked under all circumstances. Second, good work may be quickly recognized by your peers, but may take decades to move into the real world. This was certainly true here. Finally, there is value to working in a space that no one wants to touch. I have always found it more interesting, more fun, and more impactful to work in a less-crowded field (though I have also failed spectacularly by doing so).

[1]Interestingly, this paper has been cited more than 140 times as having appeared in IEEE/ACM Transactions on Networking, 1995. This is simply not true!

[1] Mohammad Alizadeh, Albert Greenberg, David A Maltz, Jitendra Padhye, Parveen Patel, Balaji Prabhakar, Sudipta Sengupta, and Murari Sridharan. 2011. Data center tcp (DCTCP). *ACM SIGCOMM computer communication review* 41, 4 (2011), 63–74.

[2] Alan Demers, Srinivasan Keshav, and Scott Shenker. 1989. Analysis and simulation of a fair queueing algorithm. In *ACM SIGCOMM Computer Communication Review*, Vol. 19. ACM, 1–12.

[3] Srinivasan Keshav. 1988. *REAL: A network simulator.* Technical Report. University of California Berkeley, Calif, USA.

[4] Srinivasan Keshav. 1994. *Packet-Pair Flow Control.* Technical Report. AT&T Bell Laboratories Technical Memorandum 11272-940927-TM.

[5] Srinivasan Keshav, Ashok K Agrawala, and Samar Singh. 1991. Design and analysis of a flow control algorithm for a network of rate allocating servers. In *Protocols for High Speed Networks II.* IFIP Press.

[6] Pratap S Khedkar and Srinivasan Keshav. 1992. Fuzzy prediction of time series. In *[1992 Proceedings] IEEE International Conference on Fuzzy Systems.* IEEE, 281–288.

[7] Naveen Kr Sharma, Ming Liu, Kishore Atreya, and Arvind Krishnamurthy. 2018. Approximating fair queueing on reconfigurable switches. In *15th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 18).* 1–16.