Reflections on "Analysis and Simulation of a Fair Queueing Algorithm"

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 It has NOT been peer reviewed.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 my personal view of the events that led to the publication of the paper "Analysis and Simulation of a Fair Queueing Algorithm" that won a Test-of-Time Award in 2007.

CCS CONCEPTS

• **Networks → Network layer protocols**.

KEYWORDS

Reminiscences, Flow Control

In this age of nearly-ubiquitous connection to the online world, with mobile devices surgically glued to the palms of the world's citizens, the state of the art in computer networking in 1988 seems quaint, if not downright primitive. End-systems were mostly dumb terminals, with 'workstations' reserved for the elite few; PCs were available but had no Internet and high-speed long-distance transmission speeds were a breathtaking 56kbps. It was in this environment that I had the great good fortune to work with Alan Demers and Scott Shenker on Fair Queueing, whose current acronym, WFQ (where W stands for weighted,) is so well-known that our work is no longer cited. Indeed, I had lunch with someone the other day who told me that hed read about WFQ in his undergrad textbook, so assumed it was something that had always existed, somehow, rather than being the output of human minds!

WFQ was born in early 1988 when Scott attended a talk by John Nagle (then at Ford Inc.) who showed that the choice of scheduling disciplines at router queues substantially influences the behaviour of a computer network and that round-robin queueing was right way to ensure that broken or malicious traffic sources could not overwhelm the rest [6]. Unfortunately, his brilliant paper, though spot on, did not work for packets of variable sizes. Scott realized this and, in summer 1988, hired me as a summer intern at Xerox PARC to address this.

At that time, Scott was new to networking, as was I, and he had ample time to mentor me, his sole intern. I recall that blissful summer of 1988 as a time when he and I would talk for what seemed like hours every day to hash out subtle details of the Fair Queueing scheme, with frequent sanity checks from Alan, who had a deeply intuitive feel for systems. My role was primarily to implement a network simulator, since Scott did not have funds to buy one! I started with the NeST simulator from Columbia [3], but by the end of summer had rewritten nearly the entire code base, and this eventually became REAL [4], which begat in turn ns, ns2, and ns3 [1, 9]. But that is another story.

Technically, Scott's main idea (and it is his entirely) was that for variable-length packets, ideal fairness is achieved by the Processor Sharing scheduling discipline, which allocates scheduling resources to head-of-line packets one infinitesimal at a time. Of course, this is impossible to carry out in practice. A feasible but very expensive approximation would be to do round-robin scheduling one bit at a time. The cost of this approximation can be dramatically reduced by doing packet-by-packet scheduling, but using an array of counters to track the ideal system's state and using these counters to decide which packet to send next. Scott's key idea was that by simulating the ideal system, the real system mimics ideal behaviour with bounded error. Indeed, this general principle can be applied to many other systems.

I implemented WFQ in my simulator and the performance results were really quite good. This encouraged us to submit a paper to SIGMETRICS 1989. I will never forget one of the reviews we got with our rejection (which I saved, so can quote verbatim *in its entirety*):

> "The title of the paper is a misrepresentation of its content. The 'algorithm' proposed is, essentially, the old heads-of-queues Processor Sharing discipline. There is no analysis beyond writing down some unlikely-looking formulae with no justification whatsoever. It is even difficult to judge the value of the simulation results, since the assumptions of the models are not clearly stated. I recommend that the paper be rejected."

After that, I did not submit a paper to SIGMETRICS for the next 25 years (for the record, my 2014 submission was also rejected).

Given the scathing reviews, we wrote and re-wrote the paper many times to polish every word and we finally submitted it to SIGCOMM 1989. It's fair to say that it attracted a lot of attention at the conference, the first SIGCOMM I attended, and the paper ended up being a classic. What we didn't fully realize at that time was that by segregating traffic streams, WFQ allows a network with cleverly-bounded sources to receive an end-to-end quality of service guarantee. This result, proved by Parekh and Gallager at MIT in 1992 [7, 8] ensured that WFQ (or its alter ego, Packet Generalized Processing Sharing, or PGPS, as they called it) became the basis for Internet QoS [2]. While that grand project never materialized, this made WFQ a technique that everyone in the networking world is familiar with.

For me, personally, our Fair Queueing paper was what Dave Clark memorably calls a 'success disaster.' Success in that due to sheer luck I was able to get a substantial boost to my research career even before I completed my PhD, with a plum job offer from Bell Labs when I graduated in 1991. But also a disaster since I've never really been able to match this prior success with my subsequent work, much as I tried. Now that I am in the final stages of my career, I'm reconciled to the fact that WFQ will be my most-cited work (my best-known work, however, is probably "How to Read a Paper" [5]).

Since those primitive days in 1988, the Internet has grown by many orders of magnitude and insinuated itself into the daily life of billion of people. Yet, the ideas in WFQ are still relevant, since they are implemented in nearly every router and switch today, and is the basis for flow separation in today's data centers. Not bad for a summer job!

REFERENCES

[1] Sandeep Bajaj, Lee Breslau, Deborah Estrin, Kevin Fall, Sally Floyd, Padma Haldar, Mark Handley, Ahmed Helmy, John Heidemann, Polly Huang, et al. 1998. *Virtual InterNetwork Testbed: Status and Research Agenda.* Technical Report 98-678. University of Southern California.

[2] David D Clark, Scott Shenker, and Lixia Zhang. 1992. Supporting real-time applications in an integrated services packet network: Architecture and mechanism. In *ACM SIGCOMM Computer Communication Review*, Vol. 22. ACM, 14–26.

[3] A Dupuy, J Schwartz, and Y Yemini. 1989. Nest: A Network Simulation Prototyping Testbed. In *1989 Winter Simulation Conference Proceedings*. IEEE, 1058–1064.

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

[5] Srinivasan Keshav. 2007. How to read a paper. *ACM SIGCOMM Computer Communication Review* 37, 3 (2007), 83–84.

[6] John Nagle. 1987. On packet switches with infinite storage. *IEEE transactions on communications* 35, 4 (1987), 435–438.

[7] Abhay K Parekh and Robert G Gallager. 1993. A generalized processor sharing approach to flow control in integrated services networks: the single-node case. *IEEE/ACM transactions on networking* 3 (1993), 344–357.

[8] Abhay K Parekh and Robert G Gallager. 1994. A generalized processor sharing approach to flow control in integrated services networks: the multiple node case. *IEEE/ACM transactions on networking* 2, 2 (1994), 137–150.

[9] George F Riley and Thomas R Henderson. 2010. The ns-3 network simulator. In *Modeling and tools for network simulation.* Springer, 15–34.