CS540 - Paper Review Report # VII

Hailu Belay Kahsay - 20155624

Title: WF^{2}Q : Worst-case Fair Weighted Fair Queueing

Author: Jon C.R. Bennett, Hui Zhang, Carnegie Mellon University

In the paper, the authors first discuss about important designing integrated services network issues: the choice of the packet service discipline at queueing points in the network. They also talk about Generalized Processor Sharing (GPS) discipline, its properties and its limitations.

The paper has two main contributions: showing the possible existence of a large discrepancy between the service provided by the packet WFQ system and the fluid GPS system, and a newly proposed packet approximation algorithm(Worst-case Fair Weighted Fair Queueing or WF2Q) that rely on services similar to those provided by GPS. The authors used Worst-case Fair Index as a metric to quantitatively measure discrepancy between the service provided by the packet WFQ system and the fluid GPS system, and they showed the effect of the discrepancy over the congestion control algorithms. After thoroughly discussing about WFQ, how it works and its limitations, the authors proposed a new and better packet approximation algorithm of GPS called Worst-case Fair Weighted Fair Queueing or WF2Q, by showing that it provides almost identical Services to GPS with a maximum difference of one packet size and it shares both the bounded-delay and fairness properties of GPS.

It is stated in the paper that the feasibility of implementing WF^{2}Q at high speeds is one of the issues that the authors didn't discuss in this paper. The authors have also talked about their future work plan: A virtual time function that lowers the complexity for the computation of virtual time function, $\mathbf{V_{Gps}()}$ and showing that the resulting discipline provides the same delay bound and has the same worst-case fair index as WF^{2}Q.

Overall, the paper presents WF^{2}Q and its key features, in detail, but issues related to the number and size of the queues are not discussed in detail? How to determine the number of queue? And what metrics should be used to set the size of each queue? Is it possible to use them for real-time application packets? Performance of WF^{2}Q with respect to packet sizing mechanism(i.e. variable Vs. fixed sized packets)?

Title: **Core-Stateless Fair Queueing: Achieving Approximately Fair Bandwidth Allocations in High Speed Networks**

Author: Ion Stoica, Scott Shenker, Hui Zhang

End-to-end congestion control solutions could be improved when routers use fair bandwidth allocation mechanisms as it protects well-behaved flows from ill-behaved ones, and allows a diverse set of end-to-end congestion control policies to co-exist in the network. Routers can implement fair bandwidth allocations by using of the following mechanisms: per-flow queueing, per-flow dropping, FIFO queuing with drop-tail. But all of these mechanisms are complex, and not cost-effectively implemented and widely deployed due to the fact that they require the routers to maintain state, manage buffers, and/or perform packet scheduling on a per flow basis.

Hence, the authors proposed a new architecture called Core-Stateless Fair Queueing that achieves reasonably fair bandwidth allocations by letting only the edge routers to maintain per-flow state not the core routers, and significantly reduces the implementation complexity. i.e. Edge routers maintain per-flow state: estimate flow rates and insert them into the packet labels. Core routers merely perform probabilistic dropping(use FIFO queuing) on input based on these labels and an estimate of the fair share rate, the computation of which requires only aggregate measurements. Packet labels are rewritten by the core routers to reflect output rates, so this approach can handle multihop situations.

The authors begun their work with two debatable assumptions: (1) fair allocation mechanisms play an important, perhaps even necessary, role in congestion control, and (2) the complexity of existing fair allocation mechanisms is a substantial hindrance to their adoption. Next, they identify and discuss about an island of routers and distinguish between the edge and the core of the island. Detail discussion regarding the solution approach: Core-stateless fair Queueing (CSFQ) and its performance, and Rate estimation algorithm and packet dropping algorithm is also included in the paper. To provide some context the authors have conducted a simulation based performance comparison between CSFQ other four algorithms: FIFO, RED, FRED and DRR.

In the paper the authors did not address the effect of large latencies and its impact on the performance of CSFQ algorithm. What will happen if we use different assumptions?