Efficient Fair Queuing using Deficit Round Robin Sriram Subramanian

The paper introduces a new Fair Queuing scheme that achieves fairness without compromising the per packet processing time (work required is O(1) per packet). The Fair Queuing algorithm that was proposed earlier required O (log n) per packet and implementing it on hardware was also expensive. Similarly fairness based on pure round robin scheduling suffers from the fact that packet lengths are not considered and approaches based on topology perform poorly.

DRR Fair Queuing derives ideas from Fair Queuing and Stochastic FQ. It uses hashing to determine the queue to which a flow has to be assigned and collisions automatically reduce the bandwidth guaranteed to the flow. Each queue is assigned a quantum and can send a packet of size that can fit in the available quantum. If not, the unused quantum gets added to this particular queue's deficit and the packet can be sent in the next round. The quantum size is a very critical parameter in the DRR scheme, determining the upper bound on the latency as well as well as the throughput. The quantum size is also central to the O(1) assurance provided by the algorithm. Only when all quantum sizes are above the maximum packet size, can this be guaranteed. The paper also formalizes two metrics - Fairness Index (measures the fairness of the queuing discipline) and Work Quotient (the effort required to process each packet), and these metrics are used to compare DRR with other queuing schemes.

Pros

- The paper uses a very simple yet elegant idea to achieve superior performance with respect to throughput fairness, and this algorithm can also be implemented in gateways in an efficient and cost effective manner.
- Through Deficit Round Robin, they provide a generic framework to implement fair queuing efficiently, and they dont advocate broad policies that would suit environments. They are careful to show the mechanism and leave the policy decisions to the implementor.
- They recognize that DRR serves well for throughput fairness, but when it comes to Latency bounds it performs rather badly. They also propose a modification to the basic algorithm that can be used to cater to real time traffic.

Cons

- Throughput takes center stage, and as explained in the paper, latency bounds are not impressive. Moreover, the method (DRR+) for handling realtime traffic doesn't work well in all cases. The realtime traffic sources have an upper bound on the amount of traffic that can be sent over a time period. But DRR tends to bunch traffic and as a result, it is possible that real time traffic could get bunched as well and in turn result in these flows being sent to the best effort queue due to the contractual violation.
- The paper also doesn't discuss the interaction and impact on congestion control mechanisms that get implemented at the source and destination. The queuing delays introduced by DRR can have interesting effects on the congestion window sizes (especially since the paper bases its calculations on a continuous backlogged flow). The experiments dont mention clearly what kind of traffic is involved - is it TCP or UDP?
- Interaction of DRR gateways with other gateways in a network - again, the bunching property of DRR gateways could result in other upstream gateways misinterpreting these as bursty traffic.
- The system depends on the Quantum size to a great extend. The paper doesn't go into the nature of these numbers, are they going to be adaptive? These gateways would also have to be tuned to suit a network, and probably they wont right out of the box.