Let it Flow: Resilient Asymmetric Load Balancing with Flowlet Switching

Erico Vanini, Rong Pan, Mohammed Alizadeh, Parvin Taheri, Tom Edsall

Presented by Fabian Rohr
Let it Flow: Resilient Asymmetric Load Balancing with Flowlet Switching
Equal Cost Multi-Path Load Balancing (ECMP): Splits traffic equally on both paths.
Congestion caused when red flow starts. ECMP does not rebalance the blue flow, instead the upper blue flow keeps congesting the marked link.
Ideal situation: Blue flow rebalanced to lower path.
Many proposed schemes available to handle such situations, like FlowBender, Hedera, Presto, CONGA. These schemes need centralized controller, switch- or end-host modifications, making them more complex. Traffic oblivious schemes are simpler, but have difficulties when used on asymmetric topologies.
Let it Flow: Resilient Asymmetric Load Balancing with Flowlet Switching
Even with symmetric layout, asymmetry occurs from link failures, congestion and degrading equipment.
Let it Flow: Resilient Asymmetric Load Balancing with Flowlet Switching
Flowlets: Groups of packets that are part of the same flow. Caused by congestion control protocol timeouts (for example when TCP waits for ACKs before continuing to send).
Let it **Flow**: Resilient Asymmetric Load Balancing with Flowlet Switching
LetFlow

How does it work?
Whenever a flowlet comes in to a switch, the switch chooses a random valid port to forward the flowlet. That’s it!
Let it Flow: Resilient Asymmetric Load Balancing with Flowlet Switching
Flowlets are elastic. A few flowlets on a high capacity path leads to longer flowlets on that path, many flowlets on a congested path lead to shorter flowlets.
Various traffic scenarios tested, all yield results similar to the picture above: ECMP degrades when load is more than 50%, since it keeps sending traffic to the congested link. LetFlow is close to CONGA’s results, which is impressive since LetFlow is a way simpler scheme than CONGA.
Simulation on large scale topology with high path asymmetry shows that LetFlow is again faster than other schemes except CONGA.
The only test where LetFlow is not beaten by CONGA: In a multi-tier topology, LetFlow performs well. CONGA is missing since it is not applicable to such topologies.
Multiple destinations: LetFlow close to optimal under high load. With low load, LetFlow does not balance perfectly, since enough capacity is available on both purple and blue paths, which leads to LetFlow not giving preference to one path or another.
In the asymmetric case, LetFlow and CONGA show the same amount of latency. In the symmetric case however, CONGA’s ability to preemptively balance the traffic yields better results than LetFlow. Due to LetFlow’s reactive approach, it will only correct once congestion has already occurred, leading to worse latency than CONGA.
Different transport protocols (DCTCP, DCQCN) have been tested as well. Result: When traffic is nicely paced, there are less opportunities for flowlets, leading to worse performance for LetFlow.
Conclusion

LetFlow is not perfect, since its reactive approach means that congestion needs to occur before LetFlow improves the situation. But it is better than ECMP, especially in asymmetric topologies, while still being a very simple scheme. The authors encourage to use this idea in other scenarios where it might yield interesting results.

On the paper

+ Nice to read
+ Simple idea
- Result-heavy paper
- An overview for CONGA would have been nice, especially since some authors were also authors on CONGA.