Valinor: Transport-agnostic packet prioritization and ordering at the edge

Erfan Sharafzadeh, Sepehr Abdous, Sougol Gheissi, Soudeh Ghorbani

Johns Hopkins University

1. Microburst Dilemmas

Datacenters are the beating heart of online communication services. Designing a datacenter network depends on many objectives;

Datacenter Network Design	Low Request Latency !
	High Application Throughput
	High Resource Utilization !
	Accomodate Bursty Traffic

However, these factors come to a contrast when facing short-term congestion periods originating from bursty traffic;

Micosecond-scale Congestion Events	Are Hard to Detect and Absorb
	Cause Short-term Queueing Delays
	Incur Costly Packet Drops

How to reconcile **low request latency** with **high resource utilization** in the face of microbursts?

2. Call for Fine-grained Load-balancing

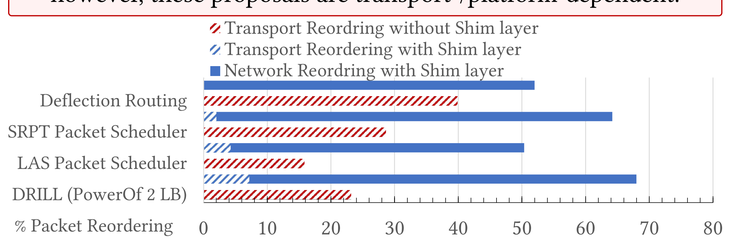
In-network Load-balancers & Schedulers operate at granularity of	Packets	Distributes the load evenly
		Prone to causing heavy re-ordering!
	Flowlets	Needs tuning to control re-ordering and load distribution
	Flows	Uneven balance for skewed workloads
		No re-ordering

If only we could get rid of packet re-ordering...

3. Generic Reordering Resiliency

Existing Solutions to Packet Reordering	Depend on transport protocol header information
	Are designed for operating system internal packet buffers

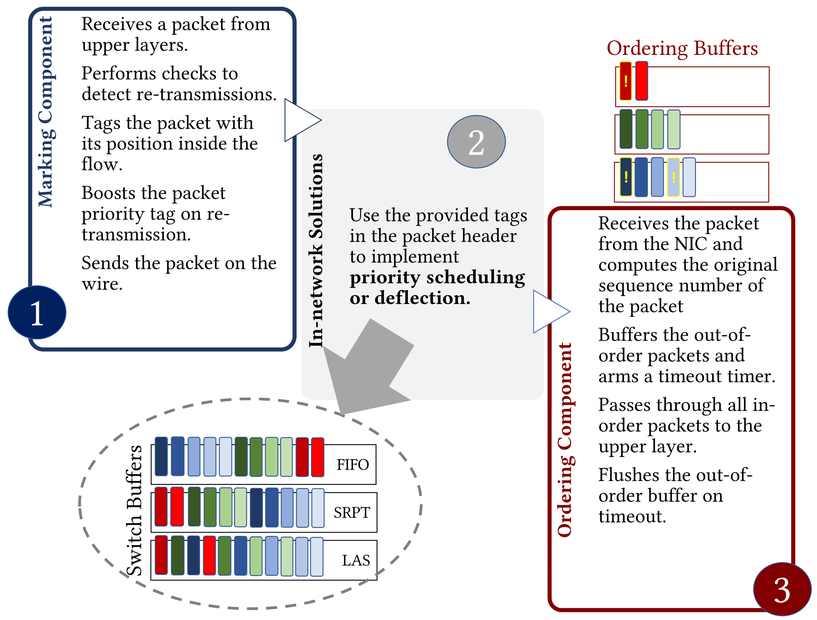
Existing fine-grained schedulers can result in severe packet re-ordering. Ordering shim layers can rearrange the packets, however, these proposals are transport-/platform-dependent.



4. Realizing Transport-agnostic Multi-discipline Scheduling and Ordering

Valinor is composed of extensions to the TX and RX datapath in data center end-hosts' network stacks and is designed with the following properties in mind:

1. Ability to handle packet reordering in a protocol-agnostic and platform-independent manner.
2. Host-centric packet prioritization to help in-network decisions.
3. Detecting and prioritizing packet re-transmissions.



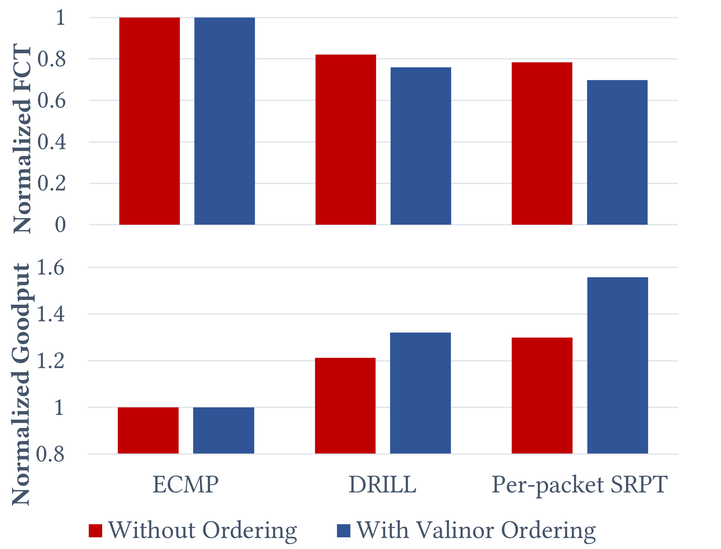
5. Implementation

User-space DPDK Implementation	Per-flow Ordering buffers in the hosts	6.5M Reqs./second throughput for UDP flows
-0.011% Goodput loss due to extra header info.	-0.003% Throughput loss due to marking	OMNET++ Implementation

Contact the first author at erfan@cs.jhu.edu

6. Preliminary Results

Valinor's packet ordering reduces the flow completion times of different load-balancing and packet scheduling schemes and substantially raises the goodput for large flows.

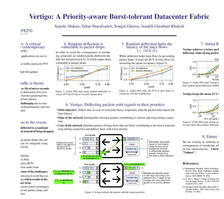


7. Ongoing Work

Each packet drop incurs an extra timeout in Valinor's ordering buffer. We're working on ways to **detect packet drops in Valinor.**

Timely reaction to microbursts requires coordinating between the network and end-hosts. We are working on a **priority deflection forwarding** mechanism called **Vertigo**.

Check out our other poster!



References

[1] Alizadeh, Mohammad, et al. "CONGA: Distributed Congestion-Aware Load Balancing for Datacenters." SIGCOMM '14, 2014.
[2] Bai, Wei, et al. "Information-Agnostic Flow Scheduling for Commodity Data Centers." NSDI '15, 2015.
[3] Geng, Yilong, et al. "Juggler: A Practical Reordering Resilient Network Stack for Datacenters." EuroSys '16, 2016.
[4] Ghorbani, Soudeh, et al. "DRILL: Micro Load Balancing for Low-Latency Data Center Networks." SIGCOMM '17, 2017.
[5] Alizadeh, Mohammad, et al. "pFabric: Minimal near-Optimal Datacenter Transport." SIGCOMM '13, 2013.