

Congestion Control for Large-Scale RDMA Deployments – Public Review

Nandita Dukkipati
Google Inc.
Mountain View, CA
nanditad@google.com

Datacenter networks are witnessing kernel bypass stacks radically different from traditional TCP/IP networking. The key drivers include cloud storage applications that require high networking bandwidth; distributed memory caches and large scale machine learning that require low latency message transfers. Widely used TCP/IP stacks do not deliver these requirements and in addition incur high CPU overhead. Kernel bypass stacks, such as RDMA over lossless Ethernet (specifically RoCEv2), can deliver low latency and high throughput along with low server CPU overhead. But just like TCP, kernel bypass transports too require well designed congestion control (CC) mechanisms so as to scale to large networks. This opportune paper takes a key step in addressing the congestion control problem for RoCE networks.

RoCE networks use Priority Flow Control (PFC) to ensure a lossless L2 network, which results in several problems including congestion spreading and unfair bandwidth allocation, making RoCE unsuitable to be used at scale. The paper presents a new CC mechanism, DCQCN, to scale RoCEv2 to large networks. DCQCN adds end-to-end CC at a per-flow granularity that causes most flows to back off before the congestion spreads. It leverages ECN marking at the switches to detect queue buildup and attempts to throttle sources before PFC is triggered, thus avoiding persistent pause messages and congestion spreading. DCQCN's rate control, implemented in the host NIC, is inspired in design by the QCN algorithm but adapted to use ECN marking without per packet ACKs. The paper addresses key practical challenges. First, ECN and PFC thresholds need to be chosen carefully in shared memory switches to ensure that ECN markings are generated prior to PFC. Second, fast convergence, fairness, and rate stability need to be balanced via a careful selection of DCQCN parameters. For this purpose, the paper develops a fluid model of DCQCN, which is used to study its dynamics. Finally the CC protocol needs to be implementable on NICs in a high-speed environment.

The reviewers appreciated three notable aspects of the work. First, the paper explores a topic of interest and provides a clear articulation of the practical chal-

lenges with RDMA deployments in datacenters. Second, it cleverly weaves bits of QCN, ECN, and PFC to come up with a CC protocol and an engineering analysis that aids algorithm tuning. Finally, the CC mechanism is implemented in Mellanox NICs, thereby facilitating real experimentation and deployment. On the flip side, the reviewers wondered on the conceptual novelty of DCQCN given that its rate control scheme is synthesized from bits of QCN and ECN. The paper compares DCQCN against a network with only PFC. There was much anticipation for a comparison against stronger candidates, such as TCP-Bolt or other high performance networking stacks. The discussion concluded that the paper tackles a timely and difficult problem, and is worth being published.

Beneath the jargon of PFC, RDMA, and RoCEv2 the paper brings up intriguing meta questions. As the paper conclusion notes, the history of CC is a tussle between responsiveness and stability. DCQCN addresses the limitations of PFC, while PFC also plays a positive role in the responsiveness of DCQCN. PFC provides a coarse grained but quick feedback to prevent packet losses just in time, making it especially useful to respond to incast congestion. The blunt response of PFC provides enough time for the slower but more fine-grained end-to-end CC to adjust sending rates and avoid persistent PFC. Could such a combination of faster and slower control loops allow for a CC design in datacenters that's fundamentally more responsive and yet stable? A second question that crops up is on lossless networks. Recently there is burgeoning work in CC, e.g. DCTCP, pFabric and FastPass, for low application latencies in data centers. This paper begs the question: should the design of large scale loss free networks join the race toward providing more stringent latencies? The paper does not solve the challenges beyond CC in making lossless networks robust, e.g., outages caused by faulty NIC spewing PFCs. The final question is whether NICs are the right place to evolve congestion control in practice, e.g., do the CPU savings outweigh the downsides such as slow iterations? The paper does not provide an answer to these questions, but certainly offers a data point toward these broader directions.