Fast Congestion Control in
RDMA-based Datacenter Networks

Jaichen Xue
Purdue University

Muhammad Usama
Chaudhry
University of Illinois at Chicago

Balajee Vamanan

T. N. Vijaykumar
Purdue University

Mithuna Thottethodi
Purdue University

CCS CONCEPTS
• Networks → Transport protocols; Data center networks;

KEYWORDS
Datacenters; RDMA; Congestion Control

1 MOTIVATION
Many modern, interactive datacenter applications have tight latency requirements due to stringent service-level agreements (e.g., under 200 ms for Web Search). TCP-based datacenter networks significantly lengthen the application latency. Remote Direct Memory Access (RDMA) substantially reduces latencies compared to TCP by bypassing the operating system via hardware support at the network interface (e.g., RDMA over InfiniBand and RDMA over Converged Ethernet (RoCE) can cut TCP’s latency by 10x [8]). As such, RDMA may soon replace TCP in datacenters.

Employing RDMA in datacenters, however, poses a challenge. RDMA provides hop-by-hop flow control and rate-based end-to-end congestion control [4]. However, RDMA’s congestion control is suboptimal for the well-known datacenter congestion problem, called incast, where multiple flows collide at a switch causing queuing delays and long latency tails [1] despite good network design [7]. Though such congestion affects only a small fraction of the flows (e.g., 0.1%), datacenter applications’ unique characteristics imply that the average latency is worsened. For example, because Web Search aggregates replies from thousands of nodes, the 99.9th percentile reply latency affects the average response time; or alternatively, dropping the slowest replies worsens the response quality. In TCP, incasts cause delays due to packet drops and re-transmissions [1]. Though the lossless RDMA does not incur packet drops, incast-induced queuing delays lengthen RDMA’s latency tail [10].

InfiniBand uses Early Congestion Notification (ECN) marks to infer imminent congestion and cuts back the sending rates [4]. While DCQCN proposes a similar scheme for RoCE, TIMELY [9] uses round-trip times (RTT) measurements, instead of ECN marks, for rate control in user-level TCP. Unfortunately, because ECN marks and RTT measurements need many round-trips to converge to the appropriate sending rates (e.g., 50 RTTs in TIMELY), the schemes are too slow for the applications’ predominantly short flows each of which lasts only a handful of round-trips. During convergence, the schemes also lose throughput due to over- and under-shooting the sending rates.

2 OUR PROPOSAL
To speed up convergence, we leverage the result in several papers and reports from large datacenter operators such as Facebook, Google and Microsoft [6]: even under typical oversubscription most congestion in datacenter networks occurs at the network edge (i.e., at the link from top-of-rack (ToR) switch to the receiver) as opposed to within the network. Our simulations confirm this result which is due to high-bandwidth network core [7] and incast at the receiver.

We make the key observation that while general congestion is complex and may require iterative convergence, the simpler and common case of receiver congestion can be addressed quicker via specialization. Without isolating this case, previous schemes apply their iterative throttling to the general case. Instead, our proposal, called Blitz, employs a divide-and-specialize approach to isolate receiver congestion and significantly speeds up the convergence. Blitz sub-divides the remaining case of in-network congestion into the simpler spatially-localized case and the harder spatially-dispersed case. For the former where the network capacity is not under pressure (e.g., due to imperfect ECMP hashing), Blitz avoids
unlike previous load-balancing schemes including TCP. Without these mechanisms, NDP would without specializing for edge congestion. (3) NUMFabric which may be unavoidable. Load balancing can alleviate local-
tion is well known, our contribution is
support packet reordering, Blitz provides hardware support
for a packet after which the packet is not deflected even at a
an alternate path is faster than being queued up in the short-
simply deflects the affected packets under the premise that
an alternate path is faster than being queued up in the short-
est path. To avoid livelock, Blitz allows only a few deflections
for a packet after which the packet is not deflected even at a
congested switch. Blitz avoids deadlocks via a widely-used
virtual-channel-based scheme [2]. Because RDMA does not
support packet reordering, Blitz provides hardware support
in the switch to keep a flow’s packets in order. While deflec-
tion is well known, our contribution is in-order flow deflection
(IOFD) unlike previous load-balancing schemes including
DIBS. As a congestion response, deflection is much lighter-
weight and quicker (well under one RTT) than rate-cutting
using iterative convergence and does not affect the sending
rates. For spatially-dispersed in-network congestion, which
is uncommon, Blitz falls back to DCQCN’s heavy-weight
rate modulation. By filtering out receiver congestion and loc-
alized in-network congestion, Blitz cuts the number of ECN
marks, which trigger DCQCN fall-backs, by 4x for typical
workloads.

3 PRELIMINARY EVALUATION

Our testbed consists of 20 nodes, each consisting of four
eight-core AMD Opteron 6320 CPUs running at 2.8 GHz and
256 GB of memory, which connect to a 36-port Mellanox
SX6025 InfiniBand switch using Mellanox ConnectX-3 Pro
HCA. The switch provides bidirectional bandwidth of 56
Gbps per port. All the nodes run RHEL6.7 (kernel version
2.6.32) and Mellanox OFED 3.3-1.0.4.

We compare the completion times of short, incast flows
and throughput of long, background flows of InfiniBand
and DASR. We initiate short 256-KB incasts from a group of
servers every 100 ms to an aggregator server. Meanwhile, we
send continuous background traffic from another server to
the aggregator. We introduce random jitter of 0–100 µs among
the incast senders in each round. While InfiniBand uses its
congestion control, we implement DASR’s rate control by
staggering the messages in time at the application layer.

Figure 1 shows the median and tail (99th percentile) flow
completion times of DASR and InfiniBand (Y-axis), for vary-
ing incast degrees (X-axis). As expected, higher incast de-

To address spatially-localized, in-network congestion, Blitz
simply deflects the affected packets under the premise that
an alternate path is faster than being queued up in the short-
est path. To avoid livelock, Blitz allows only a few deflections
for a packet after which the packet is not deflected even at a
congested switch. Blitz avoids deadlocks via a widely-used
virtual-channel-based scheme [2]. Because RDMA does not
support packet reordering, Blitz provides hardware support
in the switch to keep a flow’s packets in order. While deflec-
tion is well known, our contribution is in-order flow deflection
(IOFD) unlike previous load-balancing schemes including
DIBS. As a congestion response, deflection is much lighter-
weight and quicker (well under one RTT) than rate-cutting
using iterative convergence and does not affect the sending

REFERENCES

Padhye, Parveen Patel, Balaji Prabhakar, Sudipta Sengupta, and Murari
Fast Congestion Control in
RDMA-based Datacenter Networks


