QIONGWEN XU

qwxu@outlook.com

EDUCATION

Ph.D., Computer Science , Rutgers University, USA 2024
Dessertation: Accelerating Software Packet Processing for High-Speed Networks

M.Sc., Computer Science , Fudan University, China 2018
Dessertation: A Fast Shortest-Path Query Algorithm on Large-Scale Graphs Using Tree Decomposition

B.Eng., Electronic and Information Science and Technology , Central China Normal University, China 2015

PROFESSIONAL SUMMARY

I am currently working on the Microsoft Azure host networking team, where I design and implement high-speed, low-latency RDMA NICs that support Azure storage and AI applications. I am primarily responsible for Azure RDMA Transport (Azure MRC) and host congestion control to address challenges from AI workloads and Azure storage applications.

My expertise spans RDMA, RoCEv2, PyTorch, C/C++, and Python.

EXPERIENCE

Azure RDMA Transport (Azure MRC) desgin and implementation, Microsoft, Software Engineer 04/2025 - present
Design and develop Azure RDMA transport (Azure MRC) on Fungible DPU, which supports multiple network paths for a single RDMA Reliable Connection. Azure MRC is designed to reduce communication time of large AI model training.
- Design the Azure MRC protocol supporting RDMA Send, Write, and Read operations. The protocol consists of two layers: (1) a packet delivery layer that handles out-of-order packet arrival and packet loss, and (2) a message semantic layer responsible for in-order message completion.
- Design and implement a PoC of the Azure MRC Rx pipeline on the Fungible DPU to ensure functional correctness and high data-path performance by maximizing the use of hardware acceleration and scaling throughput across multiple processors at subflow granularity.
- Main technical stack: RDMA protocol, Azure MRC, Fungible DPU, C

RCA of PCIe backpressure to RDMA NIC, Microsoft, Software Engineer 03/2025 - present
Perform RCA of the PCIe backpressure issue on a 16-socket VHM server running SAP HANA database (i.e., slow receiver issue). The PCIe backpressure issue triggers T2 PFC watchdog in Azure, impacting all customers in the same cluster. Updating model parameters and data during AI model training can cause the same PCIe backpressure issue.
- Identify a storage test and a network test that consistently trigger PCIe backpressure, enabling reliable reproduction of the issue.
- Analyze the performance bottleneck by collecting CPU counters using EMON and NIC counters (e.g., PFC, CNP, PCIe backpressure counters).
- Figure out the host-side congestion causing PCIe backpressure: Intel UPI and HPE FLEX links have limited bandwidth, while additional traffic (e.g., multi-NUMA cache coherence data) alongside real data generates the congestion.
- Main technical stack: RDMA, CPU architecture, CPU interconnect, Intel EMON

RDMA Connection Manager, Microsoft, Software Engineer 10/2024 - 03/2025
Design and implement the client-side APIs between user-space CM and SoCMANA CM (similar to Linux kernel CM) for RDMA Reliable Connection.
- Connection establishment and release.
- Main technical stack: RDMA CM in IB spec and its implementation in MANA (Microsoft Azure Network Adapter), C

Host congestion control in RDMA network, Microsoft, Software Engineer Intern, Software Engineer 05/2023 - 03/2025
Host congestion control in RDMA network.
- Generate host congestion and test MANA's host congestion control pipeline.
- Explore and design the end-to-end system for host congestion control caused by memory bandwidth congestion.
  - Analyze why the existing RDMA congestion control algorithm (DCQCN) cannot handle host congestion and eliminate PFC.
  - The end-to-end system includes (1) monitor host congestion status; (2) monitor memory bandwidth usage of CPU-to-memory traffic and NIC-to-memory traffic; (3) allocate memory bandwidth; and (4) enforce memory bandwidth allocation.
- Main technical stack: RDMA: PFC, DCQCN, RDMA perftest; MLNX CX5, CPU architecture, CPU interconnect, Intel PCM, C++, Python

Parallelizing high-speed stateful packet processing, Rutgers University, 11/2021 - 3/2024
The objective of this project is to design a principle to parallelize software stateful network functions on multiple CPU cores in high-speed networks.
- Our experiments with realistic data center and wide-area Internet traces show that the principle we designed (state-compute replication) can scale total packet-processing throughput linearly with cores, independent of flow size distributions, across a range of realistic packet-processing programs.
- Main technical stack: stateful network functions, eBPF/XDP, parallelization, C/C++, Python
- Repo: `https://github.com/smartnic/bpf-profile`

A Reinforcement Learning-based Video Player for On-Demand Videos, Rutgers University, 09/2020 - 07/2021
Design and implement an A3C-based reinforcement learning model to predict network congestion and determine the video bitrate (resolution) to download. Engineer a video player that integrated this model.
- Develop an A3C (Asynchronous Advantage Actor Critic)-based model to predict network congestion and select video bitrate to download.
- Improve overall bitrate by 10% and eliminate all video stalls under real-world network traces.
- Main technical stack: PyTorch, Reinforcement Learning, Adaptive Video Coding

Synthesizing safe and efficient kernel extensions for packet processing, Rutgers University, 09/2019 - 10/2021
The objective of this project is to design and develop an optimizing compiler that leverages program synthesis to automatically produce safe, compact, and more performant eBPF programs.
- The optimizing compiler K2 produces code with 6–26% reduced size, 1.36–55.03% lower average packet-processing latency, and 0–4.75% higher throughput (packets per second per core) on a number of realistic benchmarks.
- K2's domain-specific techniques accelerate equivalence-checking of eBPF programs by 6 orders of magnitude.
- Main technical stack: network functions, eBPF/XDP, eBPF verifier, program synthesis, formal methods, C++
- Repo: `https://github.com/smartnic/superopt` Artifact: `https://github.com/smartnic/sigcomm21_artifact`

PUBLICATIONS

1. **Qiongwen Xu**, Sebastiano Miano, Xiangyu Gao, Tao Wang, Adithya Murugadass, Songyuan Zhang, Anirudh Sivaraman, Gianni Antichi, and Srinivas Narayana, “State-Compute Replication: Parallelizing High-Speed Stateful Packet Processing," in *Usenix Symposium on Networked Systems Design and Implementation (NSDI)*, 2025.
2. **Qiongwen Xu**, Michael Dean Wong, Tanvi Wagle, Srinivas Narayana, Anirudh Sivaraman, “Synthesizing safe and efficient kernel extensions for packet processing," in *ACM SIGCOMM*, 2021.
3. Kun Qiu, Siyuan Huang, **Qiongwen Xu**, Jin Zhao, Xin Wang, Stefano Secci, “Paracon: a parallel control plane for scaling-up path computation in SDN," in the *IEEE Transactions on Network and Service Management (TNSM)*, 2017.
4. **Qiongwen Xu**, Xu Zhang, Jin Zhao, Xin Wang and Tilman Wolf, “Fast Shortest-Path Queries on Large-Scale Graphs," in the *Proceedings of IEEE International Conference on Network Protocols (ICNP)*, 2016.

SELECTED AWARDS

First place, Nvidia North America DPU hackathon 2021
National Scholarship, Ministry of Education of China 2014, 2016