Validating TCP Behavior in DISTRI: A Comparison of Simulated and Real-World Network Performance for Distributed Computing

Imtiaz Mahmud
Lawrence Berkeley National Laboratory
Berkeley, CA, USA
imahmud@lbl.gov

Kesheng Wu
Lawrence Berkeley National Laboratory
Berkeley, CA, USA
kwu@lbl.gov

Alex Sim
Lawrence Berkeley National Laboratory
Berkeley, CA, USA
asim@lbl.gov

Anirban Mandal
RENCI, University of North Carolina
Chapel Hill, NC, USA
anirban@renci.org

Ewa Deelman
USC Information Sciences Institute
Marina del Rey, CA, USA
deelman@isi.edu

***Abstract*—Distributed computing systems require accurate network simulation tools to optimize data transfer and resource allocation across geographically distributed facilities. We extend DISTRI, a discrete-event simulator for multi-facility distributed computing, with a comprehensive Transmission Control Protocol (TCP) stack supporting multiple congestion control algorithms (CCAs) and network topologies. This unified TCP implementation enables realistic simulation of both inter-facility wide-area and intra-facility local network communications. We validate DISTRI's TCP stack accuracy by comparing inter-facility scenarios against real-world experiments on the FABRIC testbed. Using a dumbbell topology with competing data transfers, we analyze TCP Reno's behavior through congestion window (CWND) evolution, round-trip time (RTT), throughput, and packet loss patterns. Results demonstrate that DISTRI accurately captures essential TCP dynamics, with behavioral trends closely matching real-world observations. This validation establishes DISTRI as a reliable, cost-effective tool for developing and testing network optimization algorithms in distributed computing environments. Beyond qualitative validation, we outline directions for incorporating more detailed quantitative error analysis between simulated and measured TCP traces as part of future work.**

***Index Terms*—DISTRI, Distributed Computing, Network Simulation, TCP Congestion Control, Network Validation, FABRIC Testbed**

I. Introduction

Computational facilities supporting scientific research and advanced applications have grown increasingly complex, often spanning multiple geographically distributed sites with diverse internal structures [1]. These distributed computing environments enable unprecedented computational capabilities by aggregating resources across institutions, improving fault tolerance, and providing resilient infrastructure for large-scale scientific workflows [2]. As computational demands continue to surge—particularly driven by data-intensive applications in artificial intelligence [3], climate modeling, and large-scale data analytics—the ability to efficiently coordinate both inter-facility and intra-facility data transfers has become critical [4].

A fundamental challenge in these distributed environments is understanding and optimizing network performance for data communication [5]. Whether data moves between facilities across wide-area networks or within a facility's internal infrastructure, the underlying transmission control protocol (TCP) behavior critically impacts job completion times and overall system efficiency [6]. Accurately modeling these network dynamics is essential for developing effective job scheduling algorithms, optimizing resource allocation, and ensuring reliable data transfers. However, conducting extensive real-world experiments on production networks is expensive, time-consuming, and often impractical, motivating the need for accurate network simulation tools [7], [8].

DISTRI (Development and Integration of Simulation Tools for Resilient Infrastructure) [9] was developed as a discrete-event simulator for multi-facility distributed computing with agent-based job scheduling. However, the initial version lacked detailed transport-layer network simulation capabilities necessary for modeling TCP behavior in both inter-facility and intra-facility communications. In this work, we extend DISTRI [10] with a comprehensive TCP stack supporting multiple congestion control algorithms (CCAs) (Reno [11], CUBIC [12], H-TCP [13]), router functionality with active queue management (AQM) algorithms (First-In-First-Out (FIFO), Fair Queue), and multiple network topologies (mesh, dumbbell). This unified TCP implementation enables DISTRI to simulate realistic network behavior for both inter-facility wide-area data transfers and intra-facility local communications.

To validate the accuracy of our TCP implementation, we focus on inter-facility communication scenarios where network dynamics are more pronounced and challenging. We compare DISTRI's simulation behavior against real-world experiments conducted on the FABRIC testbed [14], a programmable national-scale research infrastructure. Since the same TCP stack governs both inter- and intra-facility communications

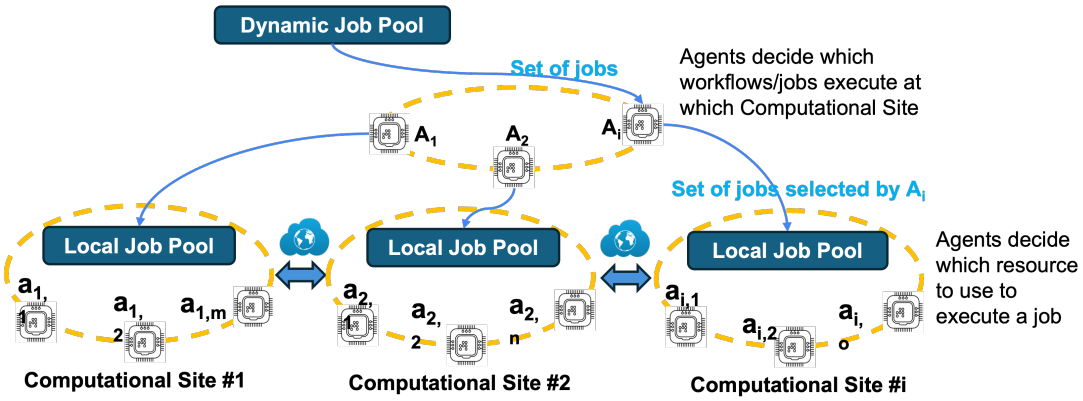

Fig. 1. DISTRI architecture showing distributed job management across multiple sites. Jobs flow from a Dynamic Job Pool to Local Job Pools at each site, where autonomous resource agents select and execute jobs based on pheromone-based load balancing.

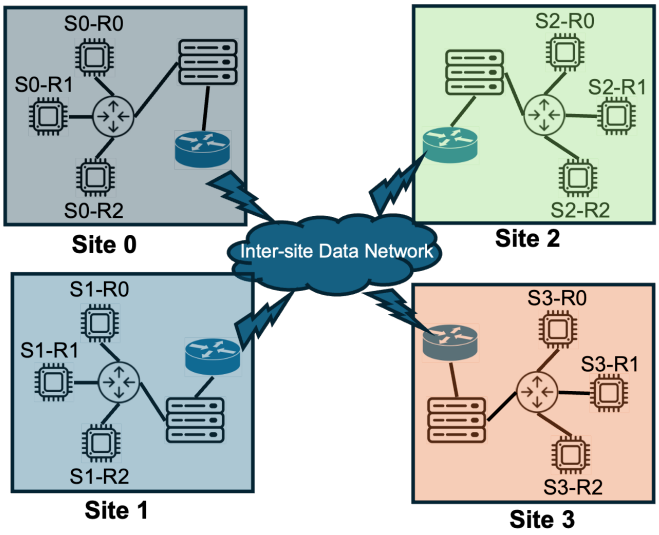

Fig. 2. DISTRI dumbbell topology with four sites. Site 0 and Site 1 host computational resources (S0-R0 and S1-R0) that simultaneously request data from Site 2 and Site 3 DTNs, respectively, creating competing flows through the shared inter-site network.

in DISTRI, validating it in the more demanding inter-facility scenario establishes confidence in its accuracy across all communication patterns.

Using a dumbbell topology with competing TCP flows, we demonstrate that DISTRI accurately captures essential TCP dynamics. Congestion window (CWND) evolution, round-trip time (RTT) variations, throughput patterns, and packet loss behaviors in DISTRI closely match FABRIC observations. While absolute values differ, the behavioral trends critical for algorithm development are remarkably similar.

The key contributions of this paper are:

- We extend DISTRI with a comprehensive TCP stack supporting multiple CCAs (Reno, CUBIC, H-TCP) and router implementations with FIFO and Fair Queue AQM algorithms, enabling realistic simulation of both inter-facility and intra-facility network communications.
- We add mesh and dumbbell network topologies to enable diverse experimental scenarios for distributed computing research.
- We validate DISTRI's TCP stack accuracy through comprehensive comparison with real-world FABRIC testbed experiments in inter-facility scenarios, analyzing CWND, RTT, throughput, and packet loss patterns.
- We demonstrate that DISTRI captures essential TCP behavioral trends across different communication patterns, establishing it as a reliable tool for developing and testing network optimization algorithms for distributed computing environments.

This validation is significant because it enables researchers to use DISTRI as a cost-effective platform for developing and testing network optimization strategies, job scheduling algorithms, and fault-tolerance mechanisms without requiring continuous access to expensive physical testbeds.

II. Related Work

Simulation tools for distributed computing and network performance analysis have been developed, but few effectively combine detailed TCP simulation with multi-facility distributed computing environments. SimGrid [7] is widely used for simulating distributed systems, and WRENCH [15] extends it for workflow management, but both lack detailed transport-layer simulation necessary for capturing TCP dynamics in inter-facility data transfers. Network simulators like NS-3 [8] provide packet-level protocol simulation but are not designed for multi-facility distributed computing environments with agent-based scheduling. Previous validation studies have compared TCP implementations [6] and analyzed congestion control behavior, with the FABRIC testbed [14] providing programmable infrastructure for controlled experiments. Network research [5] has shown that network bottlenecks significantly impact workflow completion times in distributed systems.

DISTRI [9] addresses this gap by integrating agent-based multi-facility job scheduling with comprehensive TCP stack implementation. In this paper, we validate DISTRI against

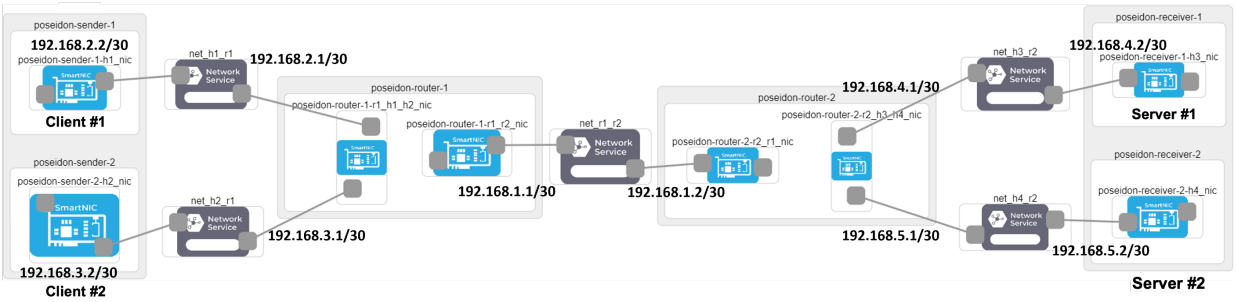

Fig. 3. FABRIC testbed topology mirroring the DISTRI setup. Two senders (Sender #1 and #2) transmit data to two receivers (Receiver #1 and #2) through a network of routers creating a bottleneck, enabling real-world validation of TCP behavior.

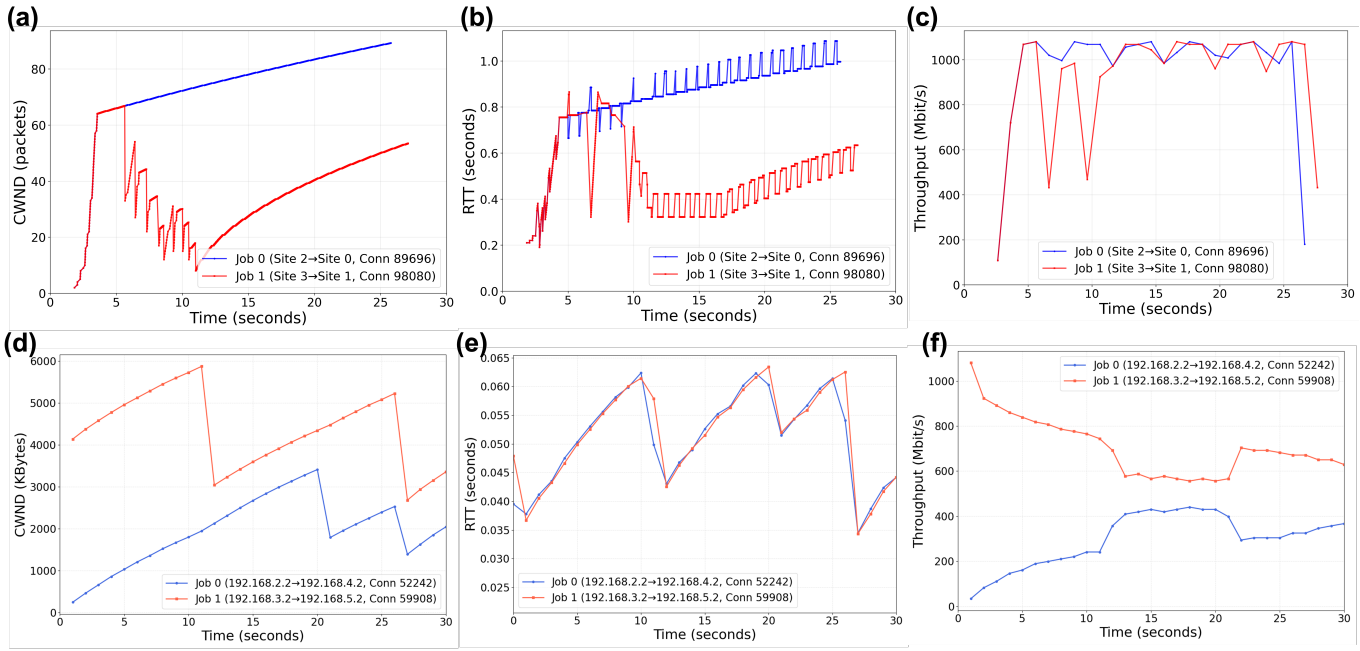

Fig. 4. Comparison of TCP behavior between DISTRI simulation (a-c) and FABRIC testbed (d-f). Top row shows DISTRI results for CWND (a), RTT (b), and throughput (c). Bottom row shows corresponding FABRIC measurements (d-f). Both environments exhibit similar behavioral trends in TCP dynamics despite differences in absolute values.

FABRIC to establish its accuracy in capturing TCP dynamics for distributed computing optimization.

III. DISTRI Overview

DISTRI is a discrete-event simulator built on SimPy [16] for modeling distributed multi-facility computing environments with agent-based resource management. The architecture, shown in Fig. 1, consists of multiple interconnected sites, each containing autonomous computational agents (resources), Data Transfer Nodes (DTNs), and a local Resource Pool that manages job assignments.

Unlike centralized schedulers, DISTRI employs a decentralized approach where resource agents independently select jobs from resource pools using swarm intelligence algorithms. This agent-based design eliminates single points of failure and improves system resilience. The extensions we present in this paper enhance DISTRI with full transport-layer networking capabilities, including TCP connection management, multiple CCAs, and router-level AQM, enabling detailed study of network performance in distributed computing scenarios.

IV. Experimental Setup

To validate DISTRI's network simulation accuracy, we designed controlled experiments comparing simulated behavior with real-world measurements from the FABRIC testbed. Both environments use a dumbbell topology configured to create competing TCP flows that share a common bottleneck link.

A. DISTRI Simulation Configuration

The DISTRI simulation, illustrated in Fig. 2, employs a four-site dumbbell topology where the inter-site data network forms a shared bottleneck. Two jobs are assigned to Site 0's resource (S0-R0) and Site 1's resource (S1-R0), which simultaneously initiate execution and request data at $t = 0$

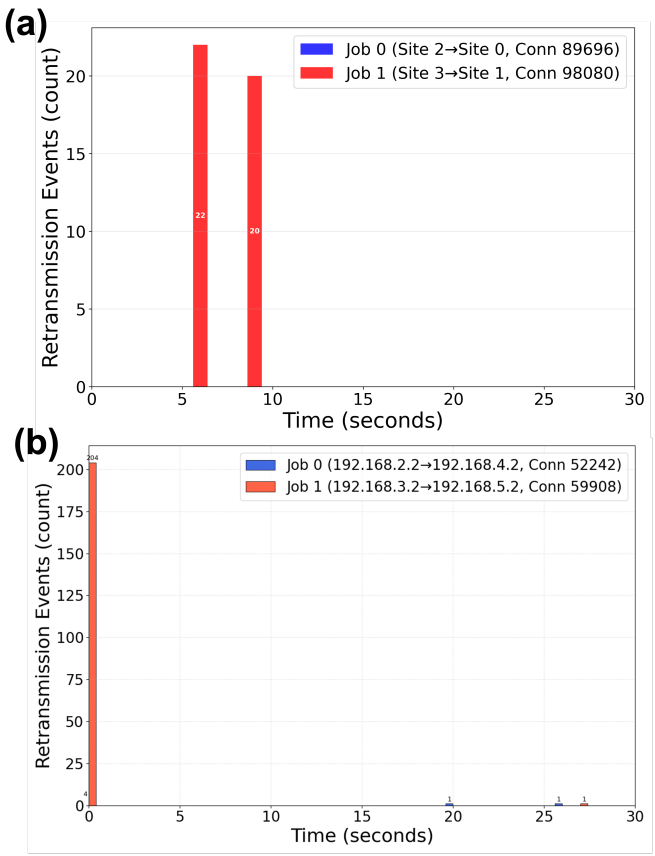

Fig. 5. Retransmission events in DISTRI (a) and FABRIC (b). Job 1 (red) experiences significantly more retransmissions than Job 0 (blue) in both environments. DISTRI shows 22 and 20 retransmission events at 6s and 9s respectively for Job 1, while FABRIC shows 204 early retransmissions for Job 1. This asymmetry directly explains the CWND suppression observed in Fig. 4.

from Site 2 and Site 3 DTNs, respectively. This creates two competing TCP flows traversing the shared bottleneck.

B. FABRIC Testbed Configuration

The FABRIC testbed experiment, shown in Fig. 3, replicates the DISTRI topology using physical infrastructure. Two senders simultaneously transfer data to two receivers through SmartNIC-equipped routers creating a controlled bottleneck matching DISTRI's configuration.

C. Network Parameters

Both environments use identical parameters: TCP Reno [11] for congestion control, FIFO queue management, 1 Bandwidth-Delay Product (BDP) queue size, 12 Mbps bottleneck bandwidth, and simultaneous flow starts. Reno was selected for its well-understood behavior, making it ideal for validation.

To facilitate reproducibility, the DISTRI simulator and the TCP validation scenarios used in this study are publicly available in the DISTRI repository [10]. The dumbbell topology experiments with TCP Reno correspond to the documented configuration that uses the –topology dumbell and –cca reno options described in the project README at [10]. TCP metrics such as congestion window, RTT, throughput, and retransmissions are exported under the directory structure in the runs/ folder, enabling readers to regenerate the plots in this paper using the same configuration.

V. Results and Analysis

Fig. 4 presents a comprehensive comparison of TCP performance metrics between DISTRI simulation and FABRIC testbed experiments. The six subplots show CWND evolution, RTT variations, and throughput patterns for both competing flows in each environment. Our analysis focuses on qualitative behavioral trends rather than absolute values, as these trends are critical for validating the simulator's ability to capture TCP dynamics.

A. Congestion Window Dynamics

The CWND evolution in Fig. 4(a,d) reveals remarkably similar patterns in both environments. Job 0 gains significant advantage with steady growth (90 packets in DISTRI, 3500 bytes in FABRIC), while Job 1 experiences severe suppression. This asymmetry is characteristic of TCP Reno's multiplicative decrease responding to packet loss—Job 1's consecutive early losses halve its CWND repeatedly, while Job 0's fewer losses enable aggressive growth. Both environments capture this fundamental Reno behavior.

B. Packet Loss Patterns

The packet loss patterns underlying the CWND behavior show similar trends in both environments. Fig. 5 quantifies these losses through retransmission events. In DISTRI, Job 1 experiences 22 retransmissions around 6 seconds and 20 around 9 seconds, causing repeated CWND reductions, while Job 0 shows no retransmissions. The FABRIC testbed exhibits even more pronounced asymmetry, with Job 1 experiencing 204 retransmissions early in the transfer while Job 0 maintains minimal losses. Despite differences in absolute retransmission counts, both environments demonstrate the same qualitative pattern: severe asymmetric packet loss that directly causes the CWND suppression and performance differences observed in Fig. 4. This correlation between packet loss and CWND evolution validates DISTRI's packet-level network simulation and queue management implementation.

C. RTT, Throughput, and Completion Patterns

RTT measurements in Fig. 4(b,e) show similar patterns reflecting queue buildup. DISTRI shows Job 0 maintaining stable RTT (0.8-1.0s) with Job 1 more variable (0.2-0.8s), while FABRIC shows comparable patterns (0.035-0.062s). Periodic spikes correspond to queue buildup followed by draining when flows back off.

Throughput in Fig. 4(c,f) reveals that despite initial asymmetry, both flows eventually achieve fair bandwidth sharing (1000-1100 Mbits/s in DISTRI, 400-600 Mbits/s in FABRIC), demonstrating Reno's additive increase mechanism balances flows despite early disparities.

In both environments, Job 0 completes earlier (25s vs 30s+ in DISTRI) due to fewer losses, validating DISTRI's ability to predict performance outcomes critical for job scheduling decisions.

D. Validation Summary

While absolute values differ between DISTRI and FABRIC due to implementation details, scaling factors, and measurement precision, the behavioral trends are remarkably consistent across all metrics. Both environments demonstrate:

- Asymmetric CWND growth driven by differential packet loss (Fig. 4)
- Quantifiable retransmission patterns showing same asymmetry (Fig. 5)
- Strong correlation between packet loss patterns and TCP performance
- RTT variations reflecting queue dynamics and congestion
- Long-term convergence to fair bandwidth sharing
- Performance-based completion time ordering

Although the absolute values of throughput, RTT, and retransmission counts differ between DISTRI and FABRIC, these discrepancies can arise from implementation details in the Linux TCP stack, measurement granularity on the testbed, modeling assumptions in the discrete-event simulator, and background system variability. Our goal in this study is therefore to validate that DISTRI reproduces the key qualitative cause-and-effect relationships in TCP behavior rather than to achieve exact packet-by-packet numerical agreement. A more systematic quantitative comparison using error metrics or correlation between simulated and measured traces is an important direction for future work that will help further calibrate the simulator. We also note that the current validation focuses on a single dumbbell topology and TCP Reno, and we plan to extend it to additional topologies and CCAs supported by DISTRI.

These consistent behavioral trends validate DISTRI's ability to capture essential TCP dynamics, making it a reliable tool for developing and testing network optimization algorithms, job scheduling strategies, and fault-tolerance mechanisms for distributed computing systems. The simulator successfully reproduces the cause-and-effect relationships between network events and TCP responses that are critical for understanding system behavior.

VI. Conclusion

We validate DISTRI's network simulation accuracy through comprehensive comparison with FABRIC testbed experiments. By extending DISTRI with a full TCP stack and router implementations, we have created a tool for distributed computing research that combines agent-based job scheduling with accurate network modeling. Our validation demonstrates that DISTRI successfully captures essential TCP dynamics—CWND evolution, RTT variations, throughput fairness, and packet loss patterns closely match real-world observations. While absolute values differ, the behavioral trends critical for algorithm development are accurately reproduced.

This establishes DISTRI as a cost-effective platform for developing network optimization strategies and testing algorithms without requiring continuous access to expensive physical testbeds. Future work will include integration of job scheduling and consensus mechanisms for multi-agent coordination in distributed computing environments. Beyond this initial validation, we plan to broaden our evaluation to additional network topologies and CCAs and to study the scalability and runtime efficiency of DISTRI for large-scale distributed workflows. We also intend to use DISTRI as a platform for investigating intelligent multi-agent coordination and workflow-aware network optimization strategies in future work.

Acknowledgment

This work was funded by the U.S. Department of Energy under the Integrated Computational and Data Infrastructure (ICDI) for Scientific Discovery, grant number DE-SC0022328. Experimental data was collected on the FABRIC testbed, which is supported by the National Science Foundation. The authors thank the FABRIC team for their support and infrastructure.

References

[1] Ewa Deelman, Tom Peterka, Ilkay Altintas, Christopher D Carothers, Kerstin Kleese van Dam, Kenneth Moreland, Manish Parashar, Lavanya Ramakrishnan, Michela Taufer, and Jeffrey Vetter. The future of scientific workflows. *The International Journal of High Performance Computing Applications*, 32(1):159–175, 2018.

[2] Patrick Kalmbach, Johannes Zerwas, Péter Babarczi, Andreas Blenk, Wolfgang Kellerer, and Stefan Schmid. Empowering self-driving networks. In *Proceedings of the afternoon workshop on self-driving networks*, pages 8–14, 2018.

[3] Shi Dong, Ping Wang, and Khushnood Abbas. A survey on deep learning and its applications. *Computer Science Review*, 40:100379, 2021.

[4] Haiwei Dong, Ali Munir, Hanine Tout, and Yashar Ganjali. Next-generation data center network enabled by machine learning: Review, challenges, and opportunities. *IEEE Access*, 9:136459–136475, 2021.

[5] Niklas Bartelheimer, Zhaobin Zhu, and Sarah Neuwirth. Automated network performance characterization for hpc systems. *International Journal of Networking and Computing*, 14(1):2–25, 2024.

[6] Imtiaz Mahmud, George Papadimitriou, Cong Wang, Mariam Kiran, Anirban Mandal, and Ewa Deelman. Elephants sharing the highway: Studying tcp fairness in large transfers over high throughput links. In *Proceedings of the SC'23 Workshops of the International Conference on High Performance Computing, Network, Storage, and Analysis*, pages 806–818, 2023.

[7] Henri Casanova. A generic framework for large-scale distributed simulations. In *Proceedings of the 10th IEEE International Symposium on Cluster Computing and the Grid*, 2001.

[8] George F Riley and Thomas R Henderson. The ns-3 network simulator. *Modeling and tools for network simulation*, pages 15–34, 2010.

[9] Imtiaz Mahmud, Pawel Zuk, Cong Wang, Mariam Kiran, Kesheng Wu, Komal Thareja, Krishnan Raghavan, Anirban Mandal, and Ewa Deelman. Distri: Development and integration of simulation tools for resilient infrastructure. In *2024 IEEE International Conference on Big Data (BigData)*. IEEE, 2024.

[10] Imtiaz Mahmud, Pawel Zuk, Cong Wang, Mariam Kiran, Kesheng Wu, Komal Thareja, Krishnan Raghavan, Anirban Mandal, and Ewa Deelman. Distri: A simulator for distributed multi-facility computing. https://github.com/swarm-workflows/DISTRI, 2024. Accessed: 2025-01-18.

[11] Van Jacobson. Congestion avoidance and control. In *Proceedings of SIGCOMM '88*, pages 314–329, Stanford, CA, USA, Aug 1988. ACM.

[12] Sangtae Ha, Injong Rhee, and Lisong Xu. Cubic: A new tcp-friendly high-speed tcp variant. *ACM SIGOPS Operating Systems Review*, 42(5):64–74, 2008.

[13] Douglas Leith and Robert Shorten. H-tcp: Tcp for high-speed and long-distance networks. In *Proceedings of PFLDnet*, volume 2004. Citeseer, 2004.

[14] Ilya Baldin, Anita Nikolich, James Griffioen, Indermohan Inder S Monga, Kuang-Ching Wang, Tom Lehman, and Paul Ruth. Fabric: A national-scale programmable experimental network infrastructure. *IEEE Internet Computing*, 23(6):38–47, 2019.

[15] Henri Casanova, Suraj Pandey, James Oeth, Ryan Tanaka, Frédéric Suter, and Rafael Ferreira Da Silva. Wrench: A framework for simulating workflow management systems. In *2018 IEEE/ACM Workflows in Support of Large-Scale Science (WORKS)*, pages 74–85. IEEE, 2018.

[16] Team SimPy. Simpy: Discrete event simulation for python. https://simpy.readthedocs.io/, 2024. Accessed: 2025-01-18.