A Study of a Deterministic Networking Framework for Latency Critical Large Scientific Data Transfers

Vijeth Kumbarahally Lakshminarayana[1], Carolina Minami Oguchi[1], Alex Sim[2], Kesheng Wu[2], Dipak Ghosal[1,2]
[1]Department of Computer Science, University of California, Davis, CA
[2]Energy Sciences Network (ESnet), Lawrence Berkeley National Laboratory, Berkeley, CA

***Abstract*—Scientific workflows often involve large data transfers, which increasingly require completion-time guarantees. To support these time-sensitive flows, the Energy Science Network (ESnet) has implemented on-demand circuits with packet priority, allowing the circuit to be utilized by other traffic when the deadline-sensitive flow is inactive. In this paper, we explore a deterministic networking framework designed to support large scientific data transfers with completion guarantees. We consider an ideal network where all nodes are time-synchronized and utilize Cyclic Queueing and Forwarding (CQF) to achieve reliable low-latency data transfers. Specifically, the CQF cycle time is configured to ensure that all data transfers between neighboring nodes are completed within the cycle time. The number of packets transferable between two neighboring nodes depends on the cycle time, propagation delay, and link bandwidth. We conduct simulations to compare the performance of the deterministic networking framework with two circuit-based schemes: one utilizing fixed bandwidth allocation for all requests and another employing dynamic bandwidth reservation, which adjusts the allocated bandwidth based on the available bandwidth along the path. Our results show that the deterministic network architecture achieves performance comparable to the dynamic bandwidth reservation scheme. We believe that a more optimized version of the time-sensitive networking protocol, exploiting multi-path routing, could offer better completion guarantees than traditional network reservation options, while enhancing the overall network bandwidth utilization.**

***Index Terms*—Scientific data transfers, Latency guarantees, Time-Sensitive Networking, Cyclic Queueing and Forwarding (CQF), Bandwidth-Reserved Circuit-Switched Routing (BRCSR), Full-Bandwidth Circuit-Switched Routing (FBCSR), Deterministic networking, Bandwidth reservation**

I. Introduction

Modern scientific endeavors in fields such as physics [1], [2], chemistry [3], biology [4], and materials science [5] are increasingly driven by vast amounts of data [6]–[8]. Many experts have termed this data-driven approach the fourth paradigm of scientific discovery [7]. This deluge of information, while promising unprecedented discoveries, poses significant challenges to traditional research paradigms [1], [2], [4]. A noticeable trend is the tighter integration between data acquisition and analysis, exemplified by high-throughput screening (HTS) in materials science [9], [10]. This integration has spurred the rapid growth of time-sensitive distributed workflows across various scientific fields [11], [12].

Currently, vast amounts of scientific data are transferred among data acquisition, storage, and analysis sites using TCP/IP networks that offer best-effort packet services [8], [13]. While this type of networking is familiar to users, it offers no guarantees on key performance metrics such as completion time [14]. Specifically, there is a non-negligible long-tail in completion time observed in applications [15]. Various approaches exist to address this performance variability. One thorough method is circuit switching, though it is no longer widely available [16]. Deployed networks primarily use packet switching, where the best-known strategy to reduce network performance variability is through reservation [17]. However, reserved network capacity is typically underutilized for other traffic, reducing overall network efficiency.

Time-sensitive networking (TSN) emerges as a promising solution to meet the growing demands for deterministic, real-time, and ultra-reliable transmission in various applications. TSN combines the flexibility of best-effort packet networks with the reliability of constant-bit-rate services [18] by establishing a contract between the network and the application [14], [19], [20]. This contract limits the transmitter of a TSN flow to a specific bandwidth, while the network reserves the necessary resources to ensure bounded latency and zero congestion loss. Additionally, TSN can sequence and deliver packets simultaneously along multiple paths, eliminating duplicates at or near their destinations.

While TSN has been shown to be particularly beneficial for applications that cannot rely on best-effort services, such as industrial control, audio and video production, and automotive control [21], [22], we believe there is significant potential for improving time-sensitive scientific workflows that are distributed across geographically distributed locations interconnected by a wide area network (WAN). These scientific workflows often transfer relatively large volumes of data among various components [23], [24]. The inherent characteristics of TSN, such as bounded latency, zero congestion loss, and deterministic data delivery, can address the constraints on the required time guarantees in these workflows.

In this paper, we explore a deterministic network architecture as a potential alternative approach to the currently deployed network reservation system known as Online Services for Circuit Provisioning and Reservation (OSCARS) [25], [26]. We consider a deterministic network that employs the principles of TSN as described in the IEEE standard. While TSN has primarily been developed for Local Area Networks (LANs), its underlying principles, including the Cyclic Queueing and Forwarding (CQF) algorithm, have been proposed for WAN contexts. However key issues such as routing, queuing, and scheduling flows on large-scale deterministic networks

based on TSN principles are largely unexplored.

In this ongoing work, we focus on an idealized network where all nodes are time-synchronized and utilize CQF to ensure reliable low-latency deterministic data transfers. Specifically, the CQF cycle time is configured so that all data transfers between neighboring nodes are completed within each cycle. The number of packets transferable between two neighboring nodes depends on the cycle time, propagation delay, and link bandwidth. Through simulation, we compare the deterministic networking framework with two circuit-based schemes: one that allocates fixed bandwidth to all requests, and another that allocates bandwidth based on the minimum available bandwidth along the path. The results indicate that, as expected, the dynamic bandwidth reservation scheme performs best. However, the deterministic network architecture delivers performance comparable to that of the dynamic bandwidth reservation scheme.

II. Current Support for Time-Sensitive Workflows

Our work is partly driven by the large scientific facilities operated by the US Department of Energy. With their sophisticated instruments and global collaborations, these facilities generate colossal datasets at unprecedented rates [2], [23], [27]. This necessitates the development of robust cross-facility workflows to manage the entire data life-cycle: from capture and storage to processing, analysis, and dissemination [5], [24], [28].

Developing and maintaining these complex workflows present significant challenges. Data must traverse multiple processing stages, requiring sophisticated coordination across diverse computing environments. Additionally, it is important to maintain flexibility in these workflows to adapt to factors like fluctuating computational demands, unexpected downtime, and the evolving needs of scientific analysis. Due to the complexity of software and tools involved, there is a strong need for automated pipelines that seamlessly connect instruments, edge computing, high-performance computing (HPC) centers, and data repositories [23], [29].

Efforts are underway to demonstrate the feasibility of automating the complex scientific workflows [2], [3], [5], [28], [30]. Many of these efforts focus on providing the flexibility of moving the data and computing for the distributed workflows. This work concerns the networking technology to improve the robustness of time-sensitive workflows which is a key workflow pattern identified in the recent surveys [23], [24].

The Online Services for Circuit Provisioning and Reservation (OSCARS) is a powerful open-source software suite developed by the Energy Sciences Network (ESnet) that empowers researchers to create, manage, and monitor dedicated network pathways for data-intensive science [25], [26]. OSCARS addresses the growing need for flexible, reliable, and high-performance network connections in scientific research. It automates the time-consuming manual process of setting up dedicated network paths.

The key feature of OSCARS is end-to-end Circuit Provisioning. Specifically, OSCARS enables researchers to define and request specific network resources, including bandwidth, latency, and path requirements. Additionally, OSCARS also supports dynamic circuit management using which users can modify, renew, or tear down existing connections in minutes, allowing for agile adaptation to evolving research needs. These and other features have allowed OSCARS to streamline scientific research across numerous disciplines by making it easier to establish and manage dedicated network connections. For instance, high-energy physics researchers can use OSCARS to move massive datasets from particle accelerators. Climate scientists benefit from OSCARS by seamlessly sharing and processing petabytes of climate model data. Similarly, in biomedical research, OSCARS accelerates the transfer and analysis of large genomic and medical imaging datasets, advancing personalized medicine and drug discovery.

III. Time-Sensitive Networks

A time-sensitive network (TSN) is a network that has strict latency and reliability requirements on data delivery. The concept is well described by N. Finn [14]. In this section, requirements and an algorithm are introduced. Time-Sensitive Networking Task Group is one of the IEEE 802.1 Working Groups [19], defining the standards of TSN including requirements, tools, and configuration. This task group was created in 2007 and was formerly known as the Audio Video Bridging (AVB) task group, but it was renamed in 2012 to the current version to follow the societal demand.

A. Requirements

The requirements for TSN are low latency with guaranteed upper bound, small variety in delivery time (small jitter), and reliable delivery. One notable characteristic of time-sensitive networks is that all nodes in the network are time-synchronized. TSN is designed mostly for the link layer (Layer 2). TSN is also categorized as a deterministic network when the path for delivery is determined before transmission, which is often the case to meet the upper bound latency guarantee requirements. Given the requirements mentioned above, the quality of the TSN is often evaluated by the number of successful time-sensitive traffic deliveries, the variance in the delivery time of time-sensitive traffic, the average time to deliver, and the throughput of non-time-sensitive traffic.

B. Cyclic Queuing and Forwarding (CQF)

One well-known TSN algorithm is Cyclic Queuing and Forwarding (CQF) [31]. It is the basic form of TSN, and it follows the architecture described above. The basic CQF has two queues in each node. When one queue is in an open state, the other queue is in a closed state, and the state switches each cycle time. The path for the traffic to be transmitted is determined by the central controller before departing. The traffic is transmitted following the cycle rhythm synchronized across the network. Once traffic leaves its source node, it travels from one node to the next node within each cycle. From the node's

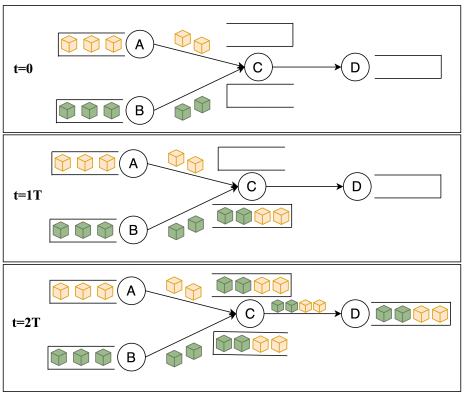

Fig. 1. The illustration of CQF algorithm. Traffic is being transmitted from two sources to a destination. In this figure, there are 2 hops from a source to the destination. T denotes a cycle time. It is illustrated that the time for traffic to get transmitted is cycle time times the number of hops, which is $T * 2 = 2T$, illustrated as $3T - 1T = 2T$ in this figure.

perspective, the traffic in the queue is dequeued and another traffic is queued into the other queue within a cycle. Therefore, in this algorithm, the time taken for a packet to reach the destination node is the cycle time times the number of hops, i.e., end-to-end packet delay = cycle time $*$ number of hops. The traffic flow of this algorithm is illustrated in Figure 1.

a) Cycle Time: The important parameter of the CQF algorithm is the cycle time denoted by T. We define the cycle time as follows. We consider the transmission of n packets each of fixed size length l bits. Between any pair of neighboring nodes $\{i, j\}$, let $T_{i,j}$ denote the minimum time that is required by node i to transmit n packets and the packets being completely received by node j. If the propagation delay of the link $\{i, j\}$ is $\delta_{i,j}$ and the data rate is $r_{i,j}$, then $T_{i,j}$ is given by

$$T_{i,j} = \frac{n \times l}{r_{i,j}} + \delta_{i,j} \tag{1}$$

The cycle time T is the defined to be the maximum $T_{i,j}$ over all pairs of neighboring node $\{i, j\}$, i.e.,

$$T = \max_{(i,j) \in S} \{T_{i,j}\} \tag{2}$$

where S is the set of all links in the network.

b) Discussion: There are important trade-offs that need to be considered in choosing a proper value of T for a network. Choosing a smaller value of n and thereby choosing a smaller cycle time T allows more fine-grained scheduling and routing of the traffic flow. However, a smaller value of T can also lead to a lower bandwidth utilization, particularly for ultra-high-speed WANs in which the transmission time of a packet is much smaller than the propagation delay. On the other hand, higher values of T will lead to high bandwidth utilization but only coarse-grained control of the traffic flow. Furthermore, in a mixed traffic environment with both best effort and time-sensitive flows, a high value of T will have high delay penalties on the best effort flows.

IV. A Comparative Analysis

The purpose of this study is to compare the performance of a WAN implementing CQF with the current OSCARS-based approach. Specifically, we consider the following network architectures to support time-sensitive flows.

1) **Bandwidth-Reserved Circuit-Switched Routing (BRCSR)**: In this scheme, the minimum bandwidth needed to guarantee that the deadline is met is allocated to each request. The paths between the source and the destination are sorted in increasing order of their distance and the shortest path that can support the bandwidth is assigned to the request. In this scheme, the bandwidth allocated to a flow does not change during the lifetime of the flow.
2) **Full-Bandwidth Circuit-Switched Routing (FBCSR)**: This scheme is similar to the BRCSR. However, if additional bandwidth becomes available, it is distributed evenly across all contending flows.
3) **Cyclic Queueing and Forwarding (CQF)**: For each link, the cycle time T determines the maximum number of packets that can be transmitted in each cycle. The node transmits an equal number of packets for each flow that uses the link. The acceptance or rejection of a flow is based on whether the link can accommodate the flow's required bandwidth while ensuring the delay bounds and not exceeding the maximum number of packets that can be transmitted.

We consider that the above three architectures are implemented in a Software Defined Network where a central controller manages data transfer requests. The workload is generated in episodes where the time between consecutive episodes is fixed and denoted by Δ. During each episode, n requests are generated. Each request i is defined by a four-tuple $\{src_i, dst_i, fsize_i, d_i\}$ where src_i and dst_i are picked randomly from the set of all nodes. The flow size $fsize_i$ and the deadline d_i determine the minimum bandwidth r_i required to meet the deadline. Specifically, $r_i = \frac{f_{size_i}}{d_i}$. The central controller admits a request if it can set up a path and allocate bandwidth on the path such that the transfer delay bound is met. If not, the request is rejected.

A. Simulation Method, Parameters, and Metrics

We implemented a discrete event simulation model using SimPy [32] and NetworkX [33] to simulate a centrally controlled network. The topology of the network is shown in Figure 2.

We only consider time-sensitive workflows in this study. The workload on the network depends on the size of each flow and the interval between consecutive requests. We employ shortest path routing for our experiment. We order possible paths and choose the shortest path that has the minimum bandwidth that is required to meet the flow deadline. In this study, we do not consider multipath routing.

We have used the following metrics to compare the three architectures.

1) Flow Latency: The total latency to transmit a flow from the source to the destination.
2) Flow Acceptance Rate: Ratio of the total number of accepted flows to the total number of requesting flows.

The simulations were run on a subsection of the ESnet topology shown in Figure 2.

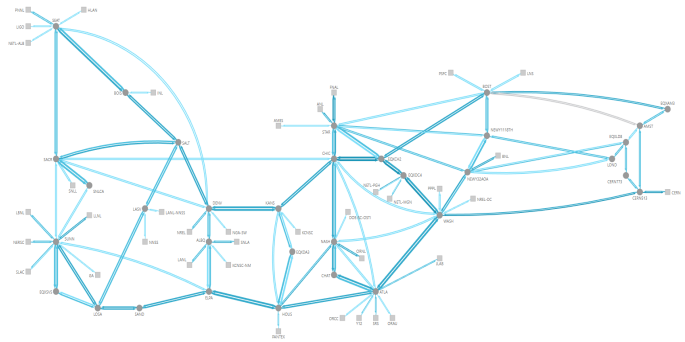

Fig. 2. The ESNet topology considered in this study. The results discussed in this paper are based on a section of the network consisting of 30 nodes and 41 link.

The network consisted of 30 nodes and 41 links. In this study, we considered the actual link propagation delays and data rates. For the workload, we considered 10 concurrent flows between randomly selected sources and destinations that arrive every 6 seconds. Each flow has a flow size that is randomly selected in the range of 700 to 850 terabytes. For each flow, the deadline was set so that a minimum of 8 Gbps was required to meet the flow deadline. For CQF, the cycle time T was varied in multiples of minimum time to transmit a packet across the link with the largest propagation delay.

B. Results and Discussion

Figure 3 shows the boxplot of flow latencies as a function of the CQF cycle time T. The minimum value of T is 6 ms.

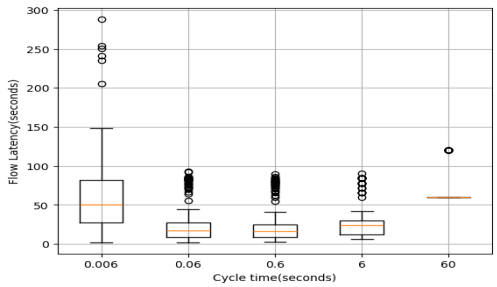

Fig. 3. The boxplot of flow latencies for different CQF cycle time T.

We have considered different values of T progressively scaled by a factor of 10 increasing to 60 seconds. The flow latencies are high for the minimum value of T, since most of the cycle time is used up by the propagation delay, resulting in lower bandwidth utilization. The latencies are also high when T is much larger than the flow latencies, such as when T is 60 seconds. The results show that a proper choice of T can optimize the average flow latency.

The flow acceptance rate as a function of T is shown in Figure 4. For CQF, the acceptance rate is low when the T is small or very high, following the trend of the flow latencies in Figure 3. Comparison with BRCSR and FBCSR indicates that with a proper choice of T, CQF can achieve comparable flow acceptance rates.

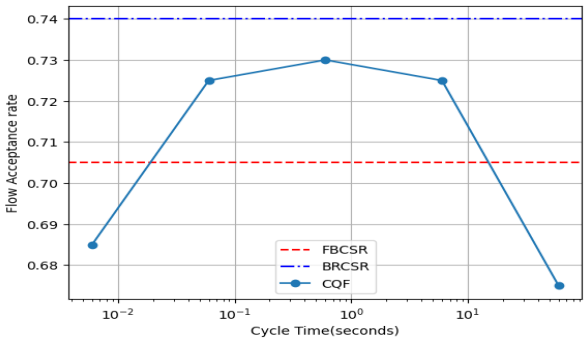

Fig. 4. Flow acceptance rate as a function of CQF cycle time T. Note that the y-axis range is cropped so the differences between the three schemes are small.

The impact of the workload on the flow acceptance rate is shown in the bar graph in Figure 5. Note that the x-axis shows

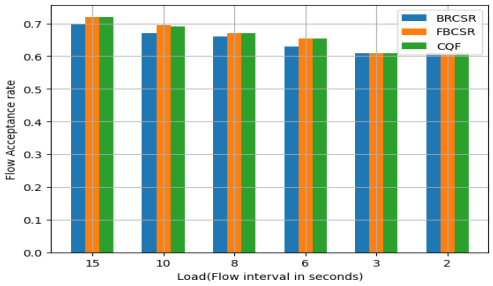

Fig. 5. Flow acceptance rate vs request load. The x-axis shows the interval between requests. Larger (smaller) intervals imply a lower (higher) request load. For CQF, T was set to 0.6 sec.

the interval between requests. Consequently, larger (smaller) intervals imply a lower (higher) load. The results show that at low load all three schemes perform the same and accept more flows. With higher load, the acceptance rate decreases, and the difference between BRCSR and the other schemes decreases due to the higher bandwidth utilization. It is worth noting that the difference between CQF and FBCSR is negligible.

The flow latency as a function of the workload is shown in Figure 6. The results show that the flow latencies of CQF and FBCSR are much better than those of BRCSR due to greater utilization of the bandwidth, as seen in Figure 6.

Overall, with the same bounded latency requirements, Cyclic Queuing and Forwarding (CQF) performs comparably to Full-Bandwidth Circuit-Switched Routing (FBCSR) in terms of flow acceptance rate and flow latency.

V. Conclusions

In this study, we conducted a comparative analysis of three strategies: Bandwidth-Reserved Circuit-Switched

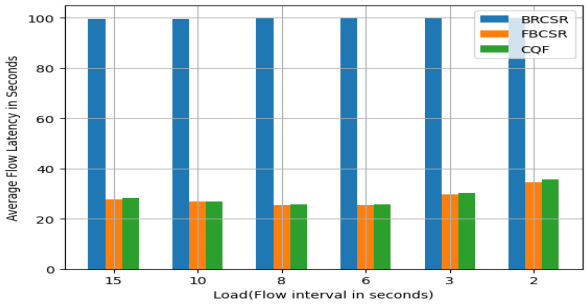

Fig. 6. Latency rate vs request load. The x-axis shows the interval between requests. Larger (smaller) intervals imply a lower (higher) request load.

Routing (BRCSR), Full-Bandwidth Circuit-Switched Routing (FBCSR), and Cyclic Queueing and Forwarding (CQF). The simulation results reveal that even the most basic implementation of CQF performs similarly to circuit-switched routing methods in terms of flow latency and flow acceptance rates.

One of the key advantages of CQF lies in the fine grained control that it provides over the network's behavior. This flexibility opens the door to further performance enhancements. Using novel optimization techniques and reinforcement learning, several network parameters can be fine-tuned to achieve performance levels that potentially exceed those of traditional circuit-switching strategies. For future research we plan to focus on exploring these optimization avenues, to fully harness the potential of deterministic networking. Such efforts could pave the way for more efficient and adaptive network routing strategies, particularly in environments that demand low latency and high reliability.

ACKNOWLEDGEMENT

This work was supported by the Office of Advanced Scientific Computing Research, Office of Science, of the US Department of Energy under Contract No. DE-AC02-05CH11231.

REFERENCES

[1] Rushil Anirudh et al. 2022 review of data-driven plasma science. *IEEE Transactions on Plasma Science*, 51(7):1750–1838, 2023.

[2] Johannes P Blaschke, Felix Wittwer, Bjoern Enders, and Debbie Bard. How a lightsource uses a supercomputer for live interactive analysis of large data sets. *Synchrotron Radiation News*, 36(4):10–16, 2023.

[3] Anees Al-Najjar et al. Enabling autonomous electron microscopy for networked computation and steering. In *2022 IEEE 18th International Conference on e-Science (e-Science)*, pages 267–277, 2022.

[4] Rafael Vescovi et al. Linking scientific instruments and computation: Patterns, technologies, and experiences. *Patterns*, 3(10), 2022.

[5] Bjoern Enders et al. Cross-facility science with the superfacility project at lbnl. In *2020 IEEE/ACM 2nd Annual Workshop on Extreme-scale Experiment-in-the-Loop Computing (XLOOP)*, pages 1–7, 2020.

[6] Tony Hey, Keith Butler, Sam Jackson, and Jeyarajan Thiyagalingam. Machine learning and big scientific data. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 378(2166):20190054, 2020.

[7] T. Hey, S. Tansley, and K. Tolle, editors. *The Fourth Paradigm: Data-Intensive Scientific Discovery*. Microsoft, October 2009.

[8] Arie Shoshani, Frank Olken, and Harry K. T. Wong. Characteristics of scientific databases. In *Proceedings of the 10th International Conference on Very Large Data Bases*, VLDB '84, pages 147–160, San Francisco, CA, USA, 1984. Morgan Kaufmann Publishers Inc.

[9] M. L. Green et al. Fulfilling the promise of the materials genome initiative with high-throughput experimental methodologies. *Applied Physics Reviews*, 4(1):011105, 03 2017.

[10] Juan J de Pablo et al. New frontiers for the materials genome initiative. *npj Computational Materials*, 5(1):41, 2019.

[11] Mark Asch et al. Big data and extreme-scale computing: Pathways to convergence-toward a shaping strategy for a future software and data ecosystem for scientific inquiry. *The International Journal of High Performance Computing Applications*, 32(4):435–479, 2018.

[12] National Science & Technology Council. Pioneering the future advanced computing ecosystem: A strategic plan. https://www.nitrd.gov/pubs/Future-Advanced-Computing-Ecosystem-Strategic-Plan-Nov-2020.pdf, 2020.

[13] Larry L Peterson and Bruce S Davie. *Computer networks: a systems approach*. Morgan Kaufmann, 2007.

[14] Norman Finn. Introduction to time-sensitive networking. *IEEE Communications Standards Magazine*, 2(2):22–28, 2018.

[15] Zhengchun Liu, Prasanna Balaprakash, Rajkumar Kettimuthu, and Ian Foster. Explaining wide area data transfer performance. In *Proceedings of the 26th International Symposium on High-Performance Parallel and Distributed Computing*, HPDC '17, pages 167–178, New York, NY, USA, 2017. Association for Computing Machinery.

[16] Sneps-Sneppe Manfred. Circuit switching versus packet switching. *International Journal of Open Information Technologies*, 3(4):27–37, 2015.

[17] Kun I Park. *QoS in packet networks*, volume 779. Springer Science & Business Media, 2004.

[18] HL Pasch and IG Niemegeers. Comparing network performance for constant and variable bit rate sources. *Computer Communications*, 16(1):27–38, 1993.

[19] Time-sensitive networking (TSN) task group. https://1.ieee802.org/tsn/. Last accessed: 2024-7-29.

[20] Mengjie Guo, Guochu Shou, Yaqiong Liu, and Yihong Hu. Software-defined time-sensitive networking for cross-domain deterministic transmission. *Electronics*, 13(7), 2024.

[21] Mahin K. Atiq et al. When ieee 802.11 and 5g meet time-sensitive networking. *IEEE Open Journal of the Industrial Electronics Society*, 3:14–36, 2022.

[22] Ahmed Nasrallah et al. Ultra-low latency (ull) networks: The ieee tsn and ietf detnet standards and related 5g ull research. *IEEE Communications Surveys & Tutorials*, 21(1):88–145, 2019.

[23] Toward a seamless integration of computing, experimental, and observational science facilities: A blueprint to accelerate discovery. Technical Report 1863562, OSTI.gov, 3 2021.

[24] Eli Dart et al. Esnet requirements review program through the iri lens: A meta-analysis of workflow patterns across doe office of science programs (final report). Technical Report 2008205, OSTI.gov, 11 2023.

[25] Chin Guok and David Robertson. ESnet on-demand secure circuits and advance reservation system (OSCARS). presented at Internet2 Joint, 2006. Additional information available at https://www.es.net/engineering-services/oscars/.

[26] Chin Guok, David Robertson, Mary Thompson, Jason Lee, and USDOE. Oscars, 6 2007.

[27] Xavier Espinal et al. The quest to solve the HL-LHC data access puzzle. volume 245, page 04027. EDP Sciences, 2020.

[28] Nicholas Tyler, Robert Knop, Deborah Bard, and Peter Nugent. Cross-facility workflows: Case studies with active experiments. In *2022 IEEE/ACM Workshop on Workflows in Support of Large-Scale Science (WORKS)*, pages 68–75, 2022.

[29] Jonathan Carter et al. Advanced research directions on ai for science, energy, and security: Report on summer 2022 workshops. Technical Report 1986455, OSTI.gov, 5 2023.

[30] Cromaz, Mario, Dart, Eli, Pouyoul, Eric, and Jansen, Gustav R. Simple and scalable streaming: The greta data pipeline*. *EPJ Web Conf.*, 251:04018, 2021.

[31] P802.1Qch – cyclic queuing and forwarding. https://1.ieee802.org/tsn/802-1qch/. Accessed: 2024-9-27.

[32] Overview — SimPy 4.1.1 documentation. https://simpy.readthedocs.io/en/latest/. Accessed: 2024-8-17.

[33] NetworkX — NetworkX documentation. https://networkx.org/. Accessed: 2024-8-17.